Bayesian Grouped Variable Selection

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Sudhir Shankar Raman
aus Chennai,Indien

Basel, 2012

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Volker Roth, Universität Basel, Dissertationsleiter
Prof. Dr. Matthais Seeger, Ecole Polytechnique Fédérale de Lausanne, Korreferent

Basel, den 24.04.2012

Prof. Dr. Martin Spiess, Dekan

Contents

Acknowledgements

This thesis has taken form and shape based on several personal and professional influences and contributions from various people who have been a part my life in these last few years. This page is an attempt to convey my gratitude to all these special people.

First and foremost, I would like to thank Volker Roth for giving me an opportunity for working with him in Basel. All the ideas presented in this thesis have developed under his keen supervision. His continuous support and insightful discussions have made this work possible. His passion as a researcher and constant pursuit for excellence has always motivated me to set high standards in the workplace. I would also like to thank Mattias Seeger for his constructive comments about the thesis which helped in improving the presentation of some parts of this work. I would like to thank Joachim Buhmann and Thomas Fuchs for their collaborations for some of this work and Peter Wild for providing data and biological insight for the biological problems that we analyzed.

I would also like to thank all the colleagues in my group for maintaining an open and casual research environment where ideas could be exchanged easily. Spending four years in Basel away from "home" would also not be possible without all the friends I have made here who have made my stay in Basel enjoyable and filled it with a lot of good memories. A special mention to the gang of "Dum Log" who have tolerated me for four years albeit with lots of complaining and bickering :).

This page cannot possibly end without giving a very special mention to the two "strong" women in my life. The first is my mother, whose sacrifices have made it possible for me to even dream of reaching this far. It was always easy to overcome frustrations in research by comparing my problems with all the problems and challenges she has faced in life. The second one is my wife, whose strength and determination have been a constant source of inspiration in more ways than she will ever realize. I will always be appreciative of her support and understanding especially in the last few months.

Last but not least, I would like to thank the financial support provided by LiverX project and the University of Basel which made this work possible.

Abstract

Traditionally, variable selection in the context of linear regression has been approached using optimization based approaches like the classical Lasso. Such methods provide a sparse point estimate with respect to regression coefficients but are unable to provide more information regarding the distribution of regression coefficients like expectation, variance estimates etc. In the recent years, there has been some progress on the Bayesian formulation for variable selection like for example, the Bayesian Lasso. Motivated by these developments, in this thesis, we build an omnibus Bayesian framework for grouped-variable selection in linear regression models. This framework is capable of summarizing the posterior distribution over the regression coefficients with estimates for the moments and the mode. The inference is carried out using Markov Chain Monte Carlo (MCMC) sampling. The estimate for the mode of the posterior distribution over regression coefficients is also generated from the same MCMC sampling algorithm with minimal changes using simulated annealing.

Going beyond simple linear regression, the framework is also extended further to accommodate generalized linear models like Poisson and binomial models with minimal changes to the framework. On the algorithm side, we develop a highly efficient MCMC sampling algorithm for inference purposes. Apart from the Poisson and binomial models, another model that has been incorporated into this framework is the Weibull model which is extensively used for survival analysis. This extension has been combined with an additional clustering component using a survival mixture-of-experts model. The clustering component is particularly useful for performing variable selection (per cluster) simultaneously with cluster identification using Dirichlet processes which avoids the need for fixing the number of clusters in advance.

The resulting framework has been applied to several biological applications like identification of novel compound bio-markers for breast cancer from tissue microarray data and analyzing splice site data for identifying distinguishing features of true splice sites. Survival data for breast cancer patients has been used to identify low-risk and high-risk patients and the significant compound markers of each group.

Notations/Abbreviations

Notations

$\mathbb{R}$	Real numbers
X	Matrix
$p(y\|a,b)$	Probability of y given parameters a and b
$\boldsymbol{x}$	A column vector
$\boldsymbol{x}^t$	Transpose of a vector x
$\propto$	Proportional to
$\sim$	Distributed as
$\boldsymbol{x}_{-i}$	A collection of all indices from vector $\boldsymbol{x}$ except i
$\circ$	Composition operator
I_d	$d \times d$ Identity matrix
K_ν	Modified Bessel function of the 2nd kind
$\mathcal{L}(\cdot)$	Likelihood function
$\mathcal{C}(\cdot)$	Cost or loss function
ℓ_p	p-norm

Abbreviations

AIC	Akaike Information Criterion
DNA	DeoxyriboNucleic Acid
DP	Dirichlet Process
EP	Expectation Propagation
GIG	Generalized Inverse Gaussian distribution
GLM	Generalized Linear Model
GLMM	Generalized Linear Mixed Model
KL	Kullback-Liebler
LARS	Least Angle Regression
LASSO	Least absolute shrinkage and selection operator
MAP	Maximum a posteriori
MCMC	Markov Chain Monte Carlo
MOE	Mixture of Experts
OLS	Ordinary Least Squares
SA	Simulated Annealing
SS	Spike and Slab
TMA	Tissue microarray

List of Figures

If a thing can be done adequately by means of one, it is superfluous to do it by means of several; for we observe that nature does not employ two instruments if one suffices.

Thomas Aquinas

1 Introduction

1.1 Data Analysis with Regression Models

In the realm of data analysis, in a large number of application domains, one frequently encounters applications where one analyzes the relationship between a large number of measurements and the response that they create. These measurements are generically referred to as input or predictor variables and the responses as response variables. More formally, we can represent this relationship based on the probability distribution $p(y|f(\boldsymbol{x}))$ where $\boldsymbol{x} \in \mathbb{R}^d$ represents a d-dimensional vector variable, $\boldsymbol{x} = (x_1, x_2, ..., x_d)$ and f is a function over $\boldsymbol{x}$ which captures the relationship between $\boldsymbol{x}$ and y and the distribution p models the stochastic nature of this relation. One of the goals of this representation is to learn the "best" function f using a learning algorithm. The optimality of the function is judged such that, if for a particular $\boldsymbol{x}$, a value of y is generated from the inferred function then it is as "close" as possible to the true value of y, also referred to as the ground truth. Such an optimal function is learned based on given pairs of n observations $(\boldsymbol{x}_i, y_i)_{i=1}^n$. Learning such relationships serves the overall purpose of being able to predict responses accurately for new inputs. Such type of learning is also known as supervised learning. Fig 1.1 illustrates the process of supervised learning.

For example, consider a biological example, where the inputs are expression values for various proteins measured from various patients who either have a particular disease or do not. Here the expression values quantify the abundance of proteins produced under a particular experimental setup. The response in this case is a binary variable which can take values $\{0, 1\}$ where 0 indicates that the patients have the particular disease and 1 indicates that the patients do not have the disease. The goal is then to learn a distribution which can take the protein expressions as input and accurately predict whether a new patient will have a disease or not. The problem is graphically described in Figure 1.2.

For analyzing such relationships, linear regression models are one of the most widely studied and used models in statistics. Due to its simplicity, it has traditionally been a popular choice for analyzing such relationships between predictor or input variables and a corresponding output or response variable. Formally, let $\boldsymbol{x}$ represent the augmented vector $\{1, x_1, x_2, ..., x_d\}$ for ease of representation and y denote a scalar response. The augmented value in $\boldsymbol{x}$ simplifies the representation of the constant term. The simplest form of a linear

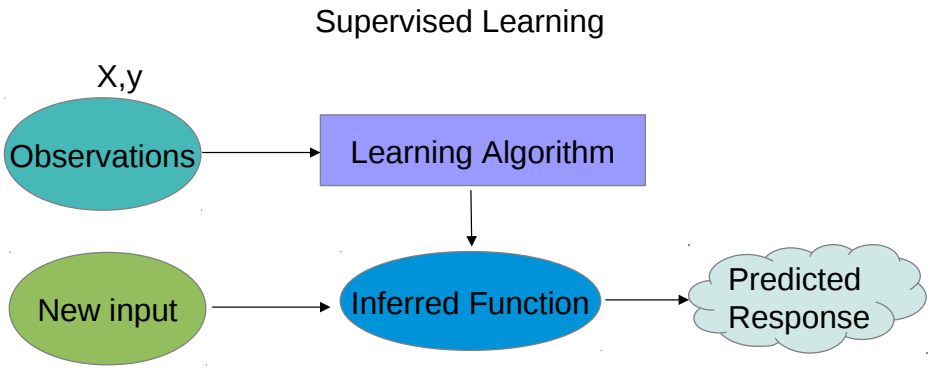


Figure 1.1.: A graphical depiction of the supervised learning problem which involves the learning of a relationship between input and response variables. The observations are used by a learning algorithm to produce a distribution which is then used for predicting responses for new inputs.

regression model is then defined as follows:

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + ... + x_d\beta_d + \epsilon = \boldsymbol{x}^t\boldsymbol{\beta} + \epsilon, \tag{1.1}$$

where ϵ represents a noise variable which models the error in the observed responses. In ordinary least squares (OLS), the error is assumed to be normally distributed with mean zero and fixed variance. Given n observations in the form of the rows in the matrix $X = \{\boldsymbol{x}_1^t, ..., \boldsymbol{x}_n^t\}$ and $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ the goal of finding the "best" function is now interpreted as finding the optimal regression coefficients $\boldsymbol{\beta}$ which minimize the disparity between predicted and observed responses which is represented in terms of a cost function. To find the optimal value of the regression coefficients, it is necessary to define a way to measure the disparity between observed and predicted responses. In the case of OLS, it is the sum of squared difference between the observed and predicted response. The resulting optimization problem is then written as:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^t\boldsymbol{\beta})^2. \tag{1.2}$$

In chapter 2, we will look at more generalized versions of the OLS model. So far we have described an optimization based view of data analysis. In section 1.2, we briefly describe an alternate view of data analysis which is probabilistic in nature.

1.2 Bayesian Inference

In the previous section, we saw how a linear regression problem was formulated as an optimization problem for finding the optimal value of the regression coefficients. This was done based on minimizing a cost function which was the sum of squared difference in the

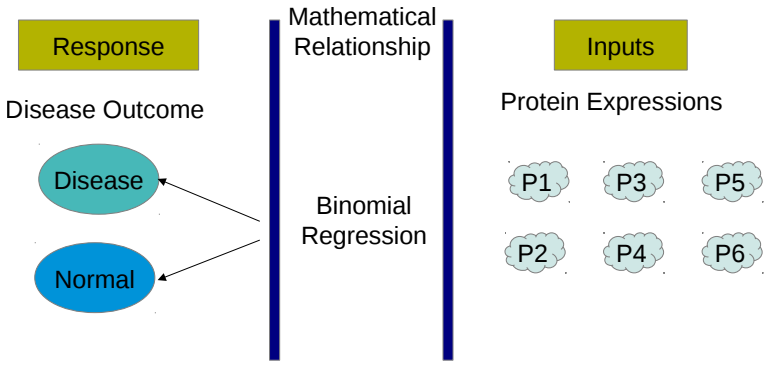


Figure 1.2.: A graphical depiction of a supervised learning problem in biology which involves characterizing the relationship between protein expression values and disease outcome. The relationship is defined using a binomial regression model.

case of OLS. Another view of the same problem is a probabilistic view. We will briefly explain the various elements of the probabilistic view in the form of Bayesian analysis and what benefits it offers.

Bayesian analysis starts with a prior belief over the parameters ($\boldsymbol{\theta}$) of a model before any data is observed which may potentially change this prior belief. This is represented in the form of a probability distribution $p(\boldsymbol{\theta})$ which encodes prior knowledge regarding the parameters. The prior distribution is very useful in channelizing data analysis in a certain direction. The second component of Bayesian analysis is the likelihood function. The observations, denoted by D, are modeled by the likelihood function, which quantifies how well the parameters explain the observed data. The goal is to model the effect of the observations on the prior belief over $\boldsymbol{\theta}$. Such an effect is modeled using Bayes theorem:

$$\begin{aligned} p(\boldsymbol{\theta}|D) &= \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \\ \Rightarrow \quad &\propto p(D|\boldsymbol{\theta})p(\boldsymbol{\theta}), \end{aligned} \tag{1.3}$$

where $p(D)$ is the normalization constant. Using Bayes' theorem, we obtain $p(\boldsymbol{\theta}|D)$, which is called as the *posterior* distribution over $\boldsymbol{\theta}$, which is so-called since it models the posterior belief in $\boldsymbol{\theta}$ based on observed data. In the regression problem defined in the previous section, the parameters are the regression coefficients $\boldsymbol{\beta}$. For the case of regression, the posterior distribution can be written as:

$$p(\boldsymbol{\beta}|\boldsymbol{y}, X) \propto p(\boldsymbol{y}|X, \boldsymbol{\beta})p(\boldsymbol{\beta}). \tag{1.4}$$

After having defined such a probabilistic interpretation of parameter learning, various quantities of interest can be learned from such a formulation. As in the optimization view, the optimal value of $\boldsymbol{\beta}$ can be found by maximizing the posterior distribution over $\boldsymbol{\beta}$.

This optimal value is known as the maximum a posteriori (MAP). This has an equivalent interpretation in the optimization based view where one seeks an optimal value. But the Bayesian formulation need not only provide a point estimate in the form of the MAP. Since the whole posterior distribution is modeled, other estimates can also be potentially generated like estimating the expectation and the variances of the regression coefficients. Hence, in summary, a Bayesian formulation is potentially beneficial since it can summarize the posterior distribution over the regression coefficients using the moments and the mode of the distribution. Information such as variances are especially useful in quantifying the uncertainty in estimates for the regression coefficients.

So far we have discussed the formulation of linear regression using both an optimization based view and a probabilistic view. We started with the intention of defining a relationship between inputs and response with the goal of accurately predicting responses for new inputs. We now look at a different goal of data analysis with respect to the input-response relationship.

1.3 Parsimony in Data Analysis

In the previous section we discussed the relationship between inputs and response with the goal of predicting responses for new inputs. But this need not be the only goal of analyzing such relationships. An alternate goal is the interpretation of the inferred relationship in terms of the significance of each input variable in predicting response. This is done with the intention of selecting a smaller subset of input variables which are considered more important than the others. Such a problem of identifying significant variables which help in characterizing the relationship between the inputs and the response is known as feature selection or variable selection. We shall call such a model a parsimonious or sparse model, since a sparse set of variables are selected.

The need for a parsimonious model via feature selection is motivated in multiple ways. The first factor influencing such a need is the possibility of existence of a natural redundancy in the underlying data. This can easily be the case in a lot of application domains where the starting point of testing a hypothesis involves looking at all possible variables that could effect the understanding of a certain hypothesis. Then, through suitable data analysis, an attempt is made to find the truly important variables which contribute to the understanding of underlying pattern in data. From an application perspective, this reduces future cost of data collection since only the significant variables identified by the data analysis step need to be measured.

But cost is not the only gain in a parsimonious model. Another very important motivating factor involves the interpretation of the input variables with respect to the real world problem being analyzed. Although a large number of variables may enhance the quality of data analysis for a particular application, such models are often too complicated to understand and interpret. Hence, it becomes useful to reduce the variable set in order to build a more understandable model of the real-world application being studied. This type of a parsimonious approach helps application domain practitioners to gain a deeper understanding of the problem being studied. This may potentially lead to designing of

novel hypotheses which are tested through further experiments. Such experiments then produce more data which may again involve a similar type of data analysis. We will call such a cycle of real-world experiments triggering data analysis and vice-versa an experimental loop. A good part of the work described in this document is motivated from this experimental loop where it becomes essential to communicate the results of data analysis in a way that it can be interpreted and utilized further for triggering further research in the particular application domain. A graphical depiction of the experimental loop is shown in Figure 1.3.

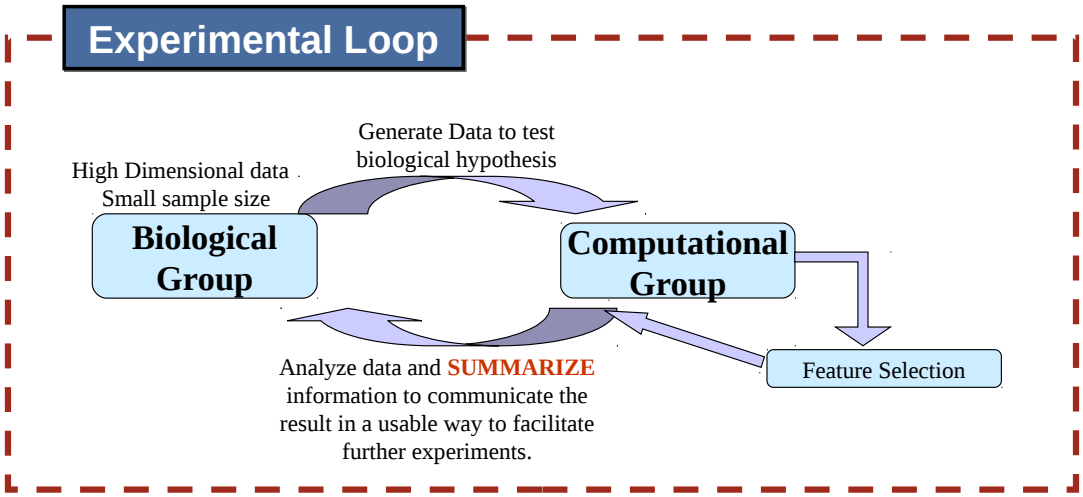


Figure 1.3.: A graphical depiction of the experimental loop which starts with data collection from a real-world application, for example analysis of data produced from biological experiments. This data is then passed on for analysis to a computational group. The analysis in this case is the generation of a sparse model. Based on the requirements, a final summary of the analysis is given back to the domain practitioners leading to further experiments which may lead to further experiments and data generation.

To explain with an example, in the biological problem of predicting disease outcome as mentioned earlier, a biologist who analyzes such a problem may not be interested only in the prediction accuracy of the model, but also in a deeper understanding of the role each protein plays on the disease outcome. The identification of a small subset of more significant proteins may trigger further analysis of the biological interpretation of how the disease is caused.

In this work, we focus on the goal of interpretation in the context of linear regression models via variable selection. We will describe some existing methods of variable selection in linear regression models and identify some of their shortcomings. We will then present our extended framework for variable selection with applications primarily in the biological domain. Although in this work, we focus on biological applications, the ideas presented

here can be applied to other application domains like image and text analysis. In the next section, we briefly describe some of the biological application areas which present the opportunity to apply variable selection methods.

1.4 Application of Sparse Models in Biology

In this section, we will give a preview of some of the application scenarios which will be used for analyzing real-world data in the context of models that we will discuss in subsequent chapters.

Protein Expressions in Tumor Analysis. A common type of data collection in biology is related to measuring protein expressions using tissue microarrays (TMA). A tissue microarray is used for the screening of genetic or protein markers across different samples as opposed to DNA (Deoxyribonucleic acid) microarrays which help in studying expressions of thousands of genes simultaneously. Tissue microarrays are often used for tumor analysis and are based on tissue or serum samples collected from patients affected by the tumor. Each array has patient specific histological samples from tumor infected tissues. The resulting TMA slides are then subjected to techniques like immuno-histochemsitry or in situ hybridization based on the specific type of analysis involved.

Due to the high throughput nature and cost-effectiveness of the tissue microarray technology, it is a preferred choice for identifying tumor related biomarkers. Biomarkers are generically used as indicators for a particular biological state. In this case, the idea is to measure protein expressions using TMAs to identify a small set of significant proteins which play an important role in tumor analysis. The identification of such proteins or biomarkers has the potential of furthering proteomics research by providing a better understanding of the underlying biological processes which result in the observed phenomena.

We shall show in subsequent chapters that the problem of biomarker detection using data collected from TMAs can be formulated as a variable selection problem in linear regression models where the variables represent the expression values of different proteins. Parsimonious models or sparse models are ideally suited for this purpose, since the goal is not only to predict a particular biological phenomena, but also to provide interpretation in terms of identifying significant proteins.

Splice Site Detection. Another type of analysis in biology is related to the DNA of an organism, which codes instructions useful for the functioning of an organism. It can be regarded as a long string of characters, where the characters are chosen from the alphabet {A,C,T,G}, like for example "ACAATTGGCTAAAAAACCGTTTGCACGA". Each character represents a particular type of nucleic acid, where A - Adenine, C-Cytosine, T- Thymine and G-Guanine. These long chains of nuclei acids are responsible for the inner workings of an organism. Within such long strings are sections known as genes which are responsible for production of proteins which in turn perform a particular function. There are two types of sub-sections within these sections which are of specific interest, namely

the exon and the intron, which alternate in a given DNA sequence. A splice site is the position(s) in the DNA which separates an intron from an exon. splice

The exons are the functional parts of the gene which are used to produce proteins. During the protein generation process, the introns are identified and discarded. In order to identify the exons and introns in a gene, a problem which is encountered is the difficulty in determining which positions are genuine splice sites. One way of tackling this problem is to infer an identification rule based on the content of the neighboring positions. A further step in this inference can be to identify which neighboring positions (or combinations of positions) are important in assessing the authenticity of a splice site. This type of analysis can then give the biologists insight into the biological reasons behind the existence of these sites and their positioning in the DNA. In later chapters, we analyze this problem using the MEMset human splice site dataset which consists of annotated splice information for a large number of sequences from the human DNA. The goal of data analysis is to use the existing annotation to detect the positions which are instrumental in distinguishing between a true and a false splice site.

Survival Analysis for Breast Cancer Patients. In biology, survival analysis problems generally involve the modeling of survival patterns of a group of patients based on a disease being analyzed. A survival pattern refers to the distribution of survival time where the meaning of "survival" can be interpreted in different ways based on the specific application. Usually, the experiment setup involves a common theme between the group of patients under consideration, for example the patients may be suffering from a particular disease and are all treated with the same medicine. In such a case survival time can be interpreted as the time till a patient does not have a re-occurrence of the disease.

The data that is collected in such a case involves survival time along with some other measurements like clinical data and gene/protein expressions. The collected data can then be analyzed in multiple ways based on the biological hypotheses being tested. The first aspect of analysis involves identifying possible sub-groups within the patient group based on the differences in their survival patterns. Identifying such differences can help in looking at the possible reasons for certain patients to have a more desirable survival pattern than some other patients and can also be useful from the perspective of personalized medicine or targeted therapies.

Another aspect of analysis can involve relating the survival patterns with the other available measurements such as clinical data and gene/protein expressions. As mentioned before, the interpretation of this relation in terms of identifying significant measurements may lead to a better understanding of the biological reasons which give rise to certain survival patterns. In this work, we look at the specific survival patterns of breast cancer patients and identify low-risk and high-risk patients through clustering. Simultaneously we identify significant proteins which can serve as bio-markers for characterizing survival patterns in patients.

1.5 Outline and Contributions

After giving a brief introduction of some of the ideas and applications related to sparse models in linear regression, we now give a more detailed roadmap of how this thesis is organized in the next few chapters. The central theme of this thesis is the description of a general framework for Bayesian grouped variable selection in the context of linear regression models. The various components of this framework are then developed which justify its generality and applicability to a variety of modeling scenarios.

In **Chapter 2**, we review some concepts and terms related to linear regression in detail and then describe some of the existing literature in the field of variable selection. The description is divided into two parts. The first part looks at an optimization based view of the problem and the second one looks at a Bayesian view which builds the motivation for our Bayesian framework for grouped-variable selection.

In **Chapter 3**, we describe the main goals of this thesis followed by a description of the Group-Lasso. Using the Group-Lasso and the Bayesian Lasso as motivation, we build a Bayesian framework for grouped variable selection which we call the Bayesian Group-Lasso. The full hierarchical model is presented along with the inference procedure in the form of Markov Chain Monte Carlo sampling. We further generalize Bayesian Group-Lasso to a variable selection framework which has an extra parameter to enforce various levels of sparsity without causing excessive global shrinkage of the regression coefficients.

In **Chapter 4**, we add a component of simulated annealing in order to provide another estimate in the form of a point estimate of the regression coefficients by estimating the mode of the posterior. This extension is formally justified by using a variational formulation approach.

In **Chapter 5**, we move beyond simple linear regression and extend the model to generalized linear mixed models so that the framework can cater to different types of data analysis problems. Two specific examples, namely the Poisson regression and binomial regression are discussed in detail with demonstrations on real-world biological applications.

In **Chapter 6**, we discuss another application, namely survival analysis and describe how it fits into the generalized linear model framework. Further, through this model, yet another extension to the framework is described by creating a clustering component for simultaneous inference of sub-groups (clusters) and respective significant features in each cluster. This is done by using a infinite mixture-of-experts model.

Finally, in **Chapter 7**, we discuss the overall thesis in the context of other parallel developments in the field of Bayesian variable selection in order to give an overall summary of these methods. We also discuss possible future work and extensions to the work described here.

The following publications have resulted out of the work presented in this thesis:

- "The Bayesian Group-Lasso for analyzing contingency tables." Sudhir Raman, Thomas Fuchs, Peter J. Wild, Edgar Dahl, Volker Roth. ICML'09: Proceedings of the 26th international conference on Machine Learning, pages 881-888, 2009.

- "Sparse Bayesian regression for grouped variables in generalized linear models." Sudhir Raman and Volker Roth. Pattern Recognition: 31st DAGM Symposium, Lecture Notes in Computer Science 5748, pages 242-251, 2009.
- "Infinite mixture-of-experts model for sparse survival regression with application to breast cancer." Sudhir Raman, Thomas J Fuchs, Peter J Wild, Edgar Dahl, Joachim M Buhmann, Volker Roth. BMC Bioinformatics 2010, 11(Suppl 8):S8 (26 October 2010).
- "MAP estimation via simulated annealing for sparse Bayesian regression using MCMC sampling." Sudhir Raman and Volker Roth, Monte Carlo Methods for Modern Applications Workshop @ NIPS 2010.

2

Variable Selection in Linear Regression Models

2.1 Introduction to Linear Regression Models

In this chapter, we lay the foundations for variable selection in linear regression models. We will first begin by describing the general setup of linear regression and then will briefly review the literature on variable selection from an optimization perspective. Finally, we will look at the Bayesian formulation of the variable selection problem in order to motivate our contributions described in the next few chapters.

A linear regression framework is defined based on the association between a d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$ known as the input or the predictor variable and a corresponding real-valued scalar $y \in \mathbb{R}$ known as the response variable. The relationship between the two variables is defined based on a linear function:

$$y = \boldsymbol{\phi}(\boldsymbol{x})^t \boldsymbol{\beta}, \tag{2.1}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is a vector of d functions $(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), ..., \phi_d(\boldsymbol{x}))$ which are known as basis functions, $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_d)$ are the parameters of the model known as regression coefficients and the use of the word "linear" indicates linearity of the function with respect to the regression coefficients. Since observed data is generally associated with noise, an error term is introduced to model the stochastic nature of the variable y:

$$y = \boldsymbol{\phi}(\boldsymbol{x})^t \boldsymbol{\beta} + \epsilon, \tag{2.2}$$

where ϵ is a random variable whose distribution is fixed based on the problem at hand. In this work, we do not consider the modeling of the distribution over the predictor variables $\boldsymbol{x}$ except when we build clustering models in the the context of survival analysis. The discussion of these models will be postponed till chapter 5. The linear regression model can be interpreted in a probabilistic framework by modeling the distribution over y:

$$y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\beta} \sim p(y|\boldsymbol{\phi}(\boldsymbol{x})^t \boldsymbol{\beta}, \boldsymbol{\theta}), \tag{2.3}$$

where $\boldsymbol{\theta}$ represents all the other parameters of this distribution which we assume to be given as of now. Assuming that we are given a set of observations $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, which

are independent and identically distributed (i.i.d), our goal is to find the value of $\boldsymbol{\beta}$ which best explains the observations in terms of the model defined above. This is done by defining a likelihood function:

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{\phi}(\boldsymbol{x}_i)^t\boldsymbol{\beta}, \boldsymbol{\theta}), \tag{2.4}$$

where the likelihood quantifies how well the data is explained based on the given parameter $\boldsymbol{\beta}$. The goal of inference is to find the parameter $\boldsymbol{\beta}$ which best explains the data or more specifically which maximizes the likelihood function. The resulting optimal value of $\boldsymbol{\beta}$, denoted by $\boldsymbol{\beta}_{ML}$, is known as the maximum likelihood estimate:

$$\boldsymbol{\beta}_{ML} = \text{argmax}_{\boldsymbol{\beta}}\ \mathcal{L}(\boldsymbol{\beta}). \tag{2.5}$$

Since the logarithm function is a monotonically increasing function of its argument, maximizing a function is equivalent to maximizing its log or minimizing the negative log. Hence, maximum likelihood estimation can be rewritten as a cost or loss minimization problem by taking negative logarithm of the likelihood:

$$\boldsymbol{\beta}_{ML} = \text{argmin}_{\boldsymbol{\beta}}\ \mathcal{C}(\boldsymbol{\beta}), \tag{2.6}$$

where $\mathcal{C}(\boldsymbol{\beta}) = -\ln(\mathcal{L}(\boldsymbol{\beta}))$ is referred to as a cost or loss function and "ln" denotes the natural logarithm function. Hence eq. (2.5) and eq. (2.6) represent two views of the same optimization problem.

Ordinary Least Squares Problem. A special case of the above defined linear regression model is the ordinary least squares (OLS), in which the basis function vector is $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$ and the error on the response variable is normally distributed, $y \sim N(y|\boldsymbol{x}^t\boldsymbol{\beta}, \sigma^2)$, where $N(\bullet|\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. From eq. (2.6), the optimization problem for OLS is written as:

$$\boldsymbol{\beta}_{ML} = \text{argmin}_{\boldsymbol{\beta}} \quad \|\boldsymbol{y} - X^t\boldsymbol{\beta}\|_2^2, \tag{2.7}$$

where $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ is a n-dimensional vector of responses and X is a $n \times d$ matrix with rows $(\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, ..., \boldsymbol{x}_n^t)$ representing the n inputs. We will refer to this loss function as the least squares loss function.

2.2 Regularization in Linear Models

In a more abstract setting, we can look at parameter estimation from the perspective of optimizing a functional in order to estimate an optimal function which minimizes a given cost functional. The space of functions over which this optimization is done is known as the hypothesis space. In the case of linear regression, the hypothesis space is the space of all linear models. In linear regression, we infer the "best" function or the optimal value for $\boldsymbol{\beta}$, by minimizing the cost function based on the available observations as given in eq. (2.6).

The objective of inferring such a function is to be able to generalize the relationship between inputs and response for unseen data so that it helps in predicting responses for new inputs where the true responses are missing. This notion of generalization can be quantified by measuring the error made in predicting responses for all unseen data. The lower the error, the better is the generalization capacity of the inferred function.

Formally, this is measured as the expected loss on the entire data space $(\boldsymbol{x}, y)$:

$$R = E_{y,\boldsymbol{x}}[\mathcal{C}(\boldsymbol{\beta})], \tag{2.8}$$

where E denotes the expectation function over the distribution $p(\boldsymbol{x}, y)$. We do not usually have access to the entire data space and only a smaller set of observations are available. Hence, in practice, we find the optimal function based on the given observations by using an approximation to the expected loss, in the form of the empirical loss function:

$$R_{emp} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{C}(\boldsymbol{\beta}). \tag{2.9}$$

To address the issue of measuring how well the inferred function generalizes over unseen data, the entire set of observations is divided into a two parts: training data and test data. Training data is used for finding the optimal value of $\boldsymbol{\beta}$ by minimizing the empirical loss function using only this data. Once the optimal value of $\boldsymbol{\beta}$ is found, the test data consisting of m observations, is used as unseen data to measure how well the inferred parameters predict the responses by again using the empirical loss function with the test data. The empirical loss for training data is known as the training error. The empirical loss for test data is known as prediction or test error and it serves as an approximation to the generalization capacity of the model.

A common problem that is faced in functional optimization is to decide how rich the hypothesis space should be in order to generalize well over unseen data. If the hypothesis space allows a very rich set of functions, it tends to generate a lower training error. Hence the resulting optimal function is said to "fit" the training data quite well. But the downside of this is the possibility of simultaneously increasing the prediction error, since the estimation is finely tuned specifically for training data. This phenomenon is known as over-fitting. The reverse problem involves choosing a very restricted hypothesis space which results in under-fitting the training data due to a restricted choice of functions. A common strategy to such problems is to build a hypothesis space in such a way that there is a balance between over-fitting and under-fitting and this process is called regularization.

In the context of linear regression models, to avoid such problems, regularization is carried out by imposing further constraints on $\boldsymbol{\beta}$. One of the most common forms of regularization in regression involves adding a constraint on the ℓ_p-norm of regression coefficients:

$$\boldsymbol{\beta}_{RL} = \text{argmin}_{\boldsymbol{\beta}}\ \mathcal{C}(\boldsymbol{\beta}) \quad \text{s.t.} \quad \|\boldsymbol{\beta}\|_p \leq \kappa, \tag{2.10}$$

where ℓ_p norm of $\boldsymbol{\beta}$ or $\|\boldsymbol{\beta}\|_p$ is defined as $(\sum_i |\beta_i|^p)^{\frac{1}{p}}$. This above form of constrained optimization can be rewritten in Lagrangian form to get an equivalent penalized version

of the optimization problem:

$$\boldsymbol{\beta}_{RL} = \text{argmin}_{\boldsymbol{\beta}} \ (\mathcal{C}(\boldsymbol{\beta}) + c\|\boldsymbol{\beta}\|_p), \tag{2.11}$$

where c is the Lagrangian parameter which tunes the amount of regularization in the model. Setting it to zero gives us back the non-regularized version of the problem which may lead to over-fitting whereas setting it to a large value can lead to under-fitting. We will use both the constrained and penalized forms interchangeably in the rest of this document. A specific case of penalized linear regression is ridge regression, which penalizes the ℓ_2 norm. For a least squares loss function as in the OLS, the ridge regression problem is written as:

$$\text{argmin}_{\boldsymbol{\beta}} \quad \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\beta}\|_2 \leq \kappa. \tag{2.12}$$

In linear regression models, the concept of regularization can be viewed in a probabilistic framework as a prior imposed on the regression coefficients $\boldsymbol{\beta}$. For the model defined in eq. (2.12), the prior over $\boldsymbol{\beta}$ is a normal distribution. The full probabilistic model can be described as:

$$\begin{aligned} \text{Likelihood: } \boldsymbol{y} &\sim N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2 I) \\ \text{Prior: } \boldsymbol{\beta} &\sim N(\boldsymbol{\beta}|\mathbf{0}, \tau^{-1}). \end{aligned} \tag{2.13}$$

Using Bayes theorem, the posterior distribution over $\boldsymbol{\beta}$ is written as:

$$p(\boldsymbol{\beta}|\boldsymbol{y}, X, \sigma^2, \tau^{-1}) \propto N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2)N(\boldsymbol{\beta}|\mathbf{0}, \tau^{-1}). \tag{2.14}$$

With the introduction of the prior distribution, the problem of maximizing the likelihood changes to the problem of maximizing the posterior distribution over regression coefficients. The resulting optimal value for $\boldsymbol{\beta}$ is known as the MAP(maximum-a-posteriori) estimate. Taking negative logarithm of eq. (2.14), we get back the penalized version of the optimization problem as in eq. (2.12). Hence the MAP estimate of $\boldsymbol{\beta}$ and the solution to eq. (2.12) are equivalent for specific values of the hyperparameters.

Likelihood Models. We now turn our attention back to the likelihood functions or equivalently the loss functions in linear regression models. In this work, we will focus on likelihood functions which are based on the exponential family of distributions. The exponential family of distributions consists of distributions which share a common form and is generically defined as:

$$f(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp[\eta(\boldsymbol{\theta})\Gamma(\boldsymbol{x}) - A(\boldsymbol{\theta})], \tag{2.15}$$

where $\boldsymbol{\theta}$ is the parameter of the distribution. This form includes most of the commonly used distributions like normal, gamma, Poisson, binomial distributions etc. The exponential family of distributions are log-concave and hence the corresponding loss functions obtained by taking negative logarithm are convex. Hence, optimizing these functions result in convex optimization problems. Also, the same holds for the regularized optimization

problems as long as the constraints result in feasible regions which are convex. For the rest of the document, while referring to likelihood or loss functions, we would implicitly assume that the likelihood function stems from the exponential family of distributions.

Generalized Linear Models. Till now, we have discussed linear regression models in the context of response variables which are real-valued scalars. To broaden the applicability of linear models to other types of response variables like binary values or count data, these models can be extended in a way that maintains the linear effect of the predictor variables. This is done by defining a generalized linear model (GLM). A GLM consists of three components, as described in [1]:

1. *Random component*: The response variable y is the stochastic component which is distributed according to some distribution with mean μ. This component is sometimes also referred to as **error structure** or **response distribution**.

2. *Systematic component*: $\eta = \boldsymbol{x}^t\boldsymbol{\beta}$ is the systematic component producing a linear predictor. So the explanatory variables $\boldsymbol{x}$ affect the response variable y, through a function of η. The two assumptions implicit in this component are the additive effects of the variables and linearity of effects.

3. *Link function*: It specifies a function which connects the mean of the distribution describing the response variable (typically an exponential family distribution) to the systematic component, as $g(\mu) = \eta$.

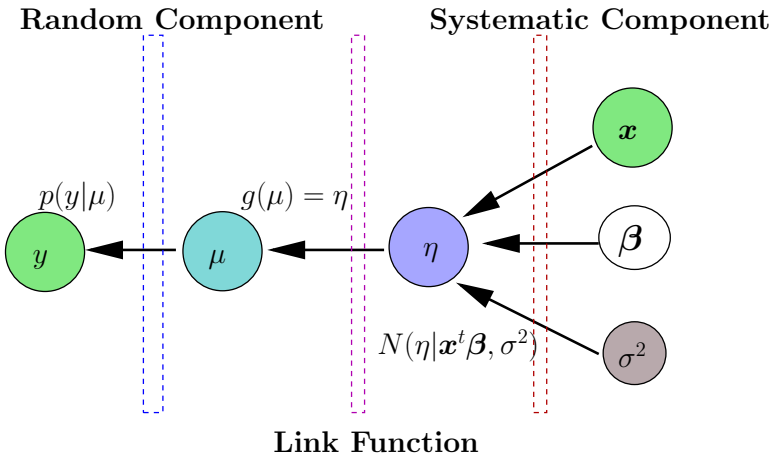


Figure 2.1.: Dependency structure of a random intercept model (i.e. a GLM with a stochastic systematic component). The dotted blocks represent the different components of the model indicating which variables are part of that block by touching the relevant connecting arrows.

Further, we can extend the standard definition of the systematic component by adding a random effect to it. This enhancement allows the linear predictor $\boldsymbol{x}^t\boldsymbol{\beta}$ to have stochastic

Table 2.1.: Commonly used likelihood functions for generalized linear models.

Response variable	**Distribution**	**Commonly Used Link Function**	$g^{-1}(\eta)$
$y = \{0, 1\}$	Bernoulli	Probit	$\Phi(\eta) =$ cdf of a Normal dist.
$y = (0, 1, 2, ...)$	Poisson	Log	$\exp(\eta)$
$y = (-\infty, +\infty)$	Gaussian	Identity	η

deviations making the model more flexible with respect to finding the effect of variables $\boldsymbol{x}$ on the response variable y. This is described as follows:

$$\eta = \boldsymbol{x}^t\boldsymbol{\beta} + \epsilon, \qquad \text{where } \epsilon \sim N(0, \sigma^2). \tag{2.16}$$

The three components of a GLM together with the random effect constitutes the simplest form of a generalized linear mixed model (GLMM) with a random effect as an intercept term, known as a random intercept model. A graphical representation of all the components of a random intercept model is shown in Figure 2.1. In the rest of the document, any reference to GLMs will be treated as a reference to the random intercept model. The full probabilistic model is written as:

$$\begin{aligned} p(y|\theta) &= h(y)\exp[y\theta - A(\theta)] \\ E(y|\theta) &= \mu \\ g(\mu) &= \eta \\ p(\eta|\boldsymbol{x}, \boldsymbol{\beta}, \sigma^2) &= N(\eta|\boldsymbol{x}^t\boldsymbol{\beta}, \sigma^2), \end{aligned} \tag{2.17}$$

where $p(y|\theta)$ is the likelihood which can be replaced by any exponential family of distributions (normal, Poisson, binomial etc.) based on the choice of modeling for the target variable y. Some examples of the commonly used distributions (which are also used in subsequent chapters) and their link functions $g(\cdot)$ are given in Table 2.1. In later chapters, we will see how the component structure of GLMs is used to extend our Bayesian framework for grouped-variable selection to specific models like Poisson and binomial model with minimal changes to the overall framework.

Towards variable selection. Although we have discussed the notion of regularization in the context of improving the generalization capability of the inferred model i.e. achieving low prediction error, there is another aspect of regression that is not suitably addressed by models like ridge regression. The other aspect of inference in linear models is the interpretation of the solution. Interpretation in this context refers to quantifying the significance of the predictor variables in predicting the response. While dealing with a large

set of predictor variables, it is often desirable to select a small subset of significant variables which have a stronger effect on the response variable and perform regression with these variables. This is especially true from an application perspective, where a domain expert may not only be interested in good prediction accuracy, but also in understanding some of the more important effects of inputs on response. The identification of such important predictor variables may lead to a further understanding of the real-world problem being analyzed. Also in keeping with Occam's razor which postulates that all things being equal, a simpler explanation is preferable to a more complex one, it is more desirable to explain the model with a smaller set of variables. The process of identifying significant variables is known as variable selection or feature selection. In linear regression models, the variable selection process involves the estimation of the regression coefficients for the significant predictor variables. This can be further interpreted as obtaining a sparse $\boldsymbol{\beta}$ vector which indicates significance of a variable x_i if the corresponding coefficient value β_i is non-zero.

For variable selection, the ℓ_2-norm regularization in linear models does not suffice since it does not encourage sparsity in the optimal values for the regression coefficients. Hence separating out the more significant variables from lesser significant ones is more difficult in this case and requires an extra selection step after obtaining the $\boldsymbol{\beta}$ estimates. In the next section, we discuss various methods that have been proposed to address the problem of variable selection.

2.3 Single Variable Selection in Linear Regression Models

First, we consider a simpler case of single variable selection, in which we assume that there is no a priori knowledge of structural associations between the predictor variables. Although there is a vast literature on variable selection, we will primarily focus on the Lasso (least absolute shrinkage and selection operator) in detail since it lays the foundation for the work that we describe in this thesis. However we will first briefly mention a couple of more methods.

Forward Selection. This method falls under the category of greedy approaches which select a subset of significant predictor variables. The method is primarily motivated by the fact that a brute force method would involve checking all the 2^d combinations of variables for judging the optimal feature-subset and the computational complexity grows exponentially with increasing d where d is the number of predictor variables. In forward selection, we start with an empty "selected-variables" set $\mathcal{S}$ which represents the variables that have been selected so far. Now, for d variables from which we have to select the relevant subset, d linear regression models are learnt containing one variable each. This is done by minimizing the loss function with only one variable in the equation. After obtaining the parameter values, the model that performs best in terms of prediction accuracy is chosen, which in turn means that the corresponding variable is chosen and added to $\mathcal{S}$.

In the next step, the same procedure is repeated by pairing the chosen variable with all the remaining $d-1$ variables. This results in adding another variable to the selected-variables set. However, since this is a greedy approach, it does not necessarily mean that

the best pair was chosen since not all $d(d-1)$ pairs were considered. This procedure continues to add features sequentially to $\mathcal{S}$ till a stopping criterion is met. The stopping condition can be, for example, a maximum number of variables to be selected or a minimum prediction accuracy to be attained. Another similar approach is backward elimination which is the reverse of forward selection. Here, we start with a full selected-variables set with all the d variables and then drop variables one by one using a similar procedure as in forward selection. Since the process of selection is greedy in nature for both approaches, it makes them less robust since the solution tends to be sub-optimal.

Non-Negative Garrote. The non-negative garrote introduced in [2] formulates the variable selection problem in the form of a two step optimization problem which tends to produce solutions that are sparse in $\boldsymbol{\beta}$. The non-zero elements of the resulting sparse $\boldsymbol{\beta}$ vector indicate that the corresponding predictor variables are selected. Hence such a formulation simultaneously infers the regression coefficient values and selects the significant variables.

The non-negative garrote is formulated as a two-step optimization problem. In the context of a least squares regression problem, the first step of the method involves solving the OLS problem in eq. (2.7) for the regression coefficients. After obtaining the OLS estimates $\hat{\boldsymbol{\beta}}^0$, the second step defines an optimization problem which selectively shrinks the OLS estimates and hence tends to produce sparse solutions in $\boldsymbol{\beta}$. The second step is specified as follows:

$$\hat{\boldsymbol{c}} = \text{argmin}_{\boldsymbol{c}} \quad \|\boldsymbol{y} - X(\hat{\boldsymbol{\beta}}^0 \circ \boldsymbol{c})\|_2^2 \quad s.t. \quad c_j \geq 0 \ \forall j, \quad \|\boldsymbol{c}\|_1 \leq \kappa, \tag{2.18}$$

where $\hat{\boldsymbol{\beta}}^0$ is the OLS estimate and the "$\circ$" operator denotes element-wise multiplication. The garrote is initialized with the OLS estimate and then shrinkage of coefficients is induced by applying an ℓ_1 norm constraint which tends to produce a sparse $\boldsymbol{c}$ vector. The final solution is $(\hat{\boldsymbol{c}} \circ \hat{\boldsymbol{\beta}}^0)$. Since the non-negative garrote depends on OLS estimates, it cannot be used for the case when $d > n$. However a modification of the non-negative garrote has been suggested in [3], where the initialization is done based on ridge regression rather than OLS in order to deal with the case of $d > n$. Since the non-negative garrote is formulated as a two-step optimization problem, it is difficult to obtain a probabilistic interpretation of the problem. We now look at a more compact representation of the variable selection problem which further motivates a Bayesian interpretation.

Lasso - Least Absolute Selection and Shrinkage Operator. Inspired by the non-negative garrote, a more compact representation of the overall optimization problem is introduced in [4] known as the Lasso. The key objective of the Lasso is the continuous shrinking of the coefficients to produce some zeroed out coefficients. Similar to the non-negative garrote, this is achieved by an optimization problem formulated as follows:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \quad s.t. \ \|\boldsymbol{\beta}\|_1 \leq \kappa, \tag{2.19}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm. Rewriting it in the Lagrangian form, it can be represented as a penalized likelihood problem:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}(\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + c\|\boldsymbol{\beta}\|_1), \tag{2.20}$$

where c represents the Lagrange parameter. The Lasso is also a special case of the more generalized penalized regression problem also called as bridge regression introduced by [5]:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}(\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + c\|\boldsymbol{\beta}\|_q), \tag{2.21}$$

where $q \geq 0$. The special case of $q = 1$ represents the Lasso and $q = 2$ is the ridge regression as defined in eq. (2.12). Another similar model known as the basis pursuit (see [6]) addresses the problem of overcomplete representations, or in other words cases where the number of basis functions exceeds the number of samples. It is almost identical to the Lasso, the only difference being that the loss function and the constraint are reversed:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}\|\boldsymbol{\beta}\|_1 \quad s.t.\ \boldsymbol{y} = X\boldsymbol{\beta}. \tag{2.22}$$

More generally, we can formulate the Lasso for a generic set of likelihood functions:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}(\mathcal{C}(\boldsymbol{\beta}) + c\|\boldsymbol{\beta}\|_1). \tag{2.23}$$

Since we are considering only the exponential family of likelihood functions, the Lasso formulation is a convex optimization problem which, below a certain threshold of κ, has a tendency to approach a solution $\hat{\boldsymbol{\beta}}$ which consists of some exact zeros and hence is a sparse solution. As in the non-negative garrote, the sparse nature of the solution serves the dual purpose of estimating the coefficients and also performing variable selection, where the variables corresponding to the non-zero coefficients are the ones which are "selected". Based on eq. (2.21), we also notice that the Lasso ($q = 1$) is the threshold for q, below which the problem becomes non-convex. This is due to the fact that all ℓ_q norms with $q \geq 1$ are convex functions and for $q < 1$ these norms are semi-norms and violate the triangle inequality and hence are non-convex regions. Hence bridge regression is convex for $q \geq 1$ and non-convex for $q < 1$. The Lasso ($q = 1$) has been very popular since it can be solved using convex optimization techniques without running into issues of local minima. A particularly fast implementation is available in the form of least-angle regression (LARS package in R), see [7]. The motivation behind using the Lasso for a Bayesian interpretation also follows from its compact representation which is easily formulated in a probabilistic setting as a product of likelihood and prior.

Model Selection. The Lasso formulation has an extra parameter c which is the Lagrangian parameter. Till now we considered solving the Lasso problem assuming c to be fixed. Fixing c to different values gives rise to different models. Hence c can be viewed as a model selection parameter which also needs to be learnt as a part of the inference procedure. Henceforth, we will refer to it interchangeably as a model selection or Lagrangian parameter. This parameter is usually learned via cross-validation. In this procedure, the

training data is divided into two parts. One part is used to train the model with a fixed value of c and the other part is used for calculating the prediction accuracy of the model. For each c, this procedure is averaged out for different divisions of the training data. After doing this for a range of values for c, the value that gives least prediction error is chosen and the full training data is then used to obtain the final Lasso estimates. For different values of c, the resulting Lasso estimates are plotted in terms of **solution paths** which plot how the value of each regression coefficient evolves with the changing values of c. Each path represents one regression coefficient. An example of a solution path plot is shown in Figure 2.2. The least-angle regression implementation for the Lasso in [7] exploits the

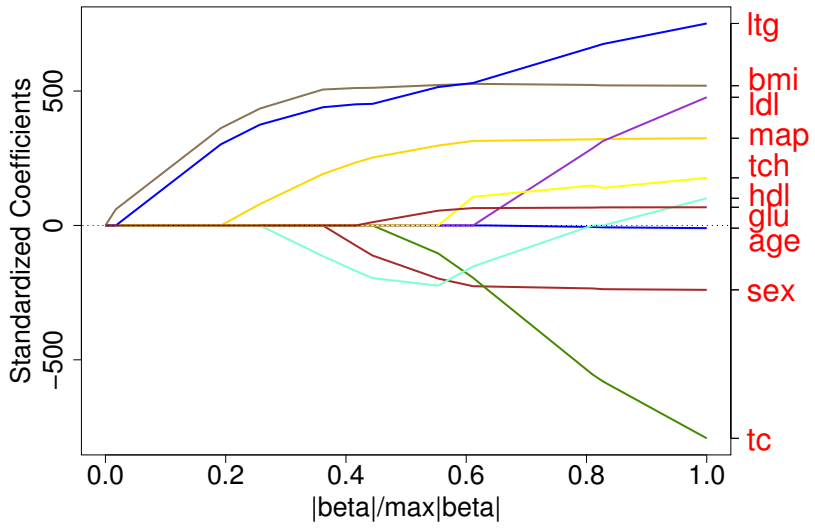


Figure 2.2.: Plot of the Lasso solution path generated with diabetes dataset (see [7]) using the LARS R package which contains a standard Lasso implementation. Each path represents the trace of the values taken by a particular coefficient for increasing values of κ.

fact that the solutions paths for the Lasso are piece-wise linear. This results in significant computational gains as the solution paths can be computed very efficiently.

Standard Error Estimates. As discussed in [8], since the Lasso is non-differentiable, it is difficult to get an estimate for the standard error of the regression coefficient estimates. This is due to the fact that the Hessian is not defined at the optimal solution. An approximation to the covariance matrix of the coefficient estimates from the Lasso has been suggested in [4]. However this approximation works only for the non-zero coefficients. For the zero coefficients, the standard error is estimated to be zero. A better estimate was provided in [9] which worked for the zero coefficients as well but only in the case of $d < n$. Bootstrapping is another alternative method for estimating standard error. But as discussed in [8], the bootstrap estimates for the Lasso are not consistent for the zero coefficients. One of the advantages of the Bayesian framework that we present in this work

is that we are able to summarize the distribution over regression coefficients with estimates for the moments and the mode by obtaining samples which closely resemble samples from the distribution over regression coefficients.

Another issue with the Lasso is that whenever there exists a group of significant and highly correlated predictor variables, the Lasso tends to select only one variable from the group, since it is redundant to also select all the other variables. In the extreme case of the variables having an exact linear relationship with each other, the variable is randomly chosen from the group. To tackle this issue, another formulation for variable selection via the elastic net has been proposed in [10]. In this work, we see how a Bayesian approach to the problem resolves this issue regarding correlated predictor variables.

2.4 Grouped-Variable Selection

In this section, we will introduce grouped-variable selection briefly and a more detailed description will be given in the next chapter. Although the mechanism for variable selection is introduced via the Lasso, it is still insufficient for problems where the predictor variables have a predefined layer of structural associations which introduces further constraints in the variable selection process. An example of such structural associations is a group structure where the predictor variables are divided into groups and the selection problem involves selecting whole groups of variables rather than individual variables. Hence in the context of regression, the desired solution requires entire groups of related coefficients to be selected (non-zero) or entire groups to be zeroed-out indicating non-selection.

An example of such a group structure which arises naturally is while regressing predictor variables which are categorical in nature. The categorical variables are expressed as groups of dummy variables and hence the original problem of selecting significant categorical variables transforms into the problem of selecting groups of dummy variables, each group representing a single categorical variable. Another example of a group structure in regression is the k-th order polynomial expansions of the predictor variables where the groups consist of products over combinations of variables up to degree k.

Motivated by these modeling scenarios, a grouped variation of the Lasso, i.e. Group-Lasso, is introduced in [11]. The modified least squares penalized optimization problem is formulated as follows:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}(\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + c\sum_{g=1}^{G} \|\boldsymbol{\beta}_g\|_2), \tag{2.24}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm, G is the number of groups and $\boldsymbol{\beta}_g$ is a sub-vector of $\boldsymbol{\beta}$ which represents all the regression coefficients of group g. The key modification lies in the penalization which involves an ℓ_1-ℓ_2 constraint on the regression coefficients. This penalty encourages sparsity of $\boldsymbol{\beta}$ at the level of groups which is represented by the ℓ_1 norm between groups and within groups there is an ℓ_2 norm. The above formulation is again a convex optimization problem.

Since the Lasso solution path was piece-wise linear, it was possible to design a proce-

dure to efficiently compute all the solutions paths as demonstrated in [7]. However, the Group-Lasso solution path is, in general, not piece-wise linear, hence it requires intensive computation for large-scale problems. A fast active-set algorithm was proposed in [12] to deal with large scale problems. The issue related to non-uniqueness of solutions in Group-Lasso problems is identified in [12] and suitable test is defined for verifying if the solution for the given Group-Lasso problem is unique or not. Also the Group-Lasso has been extended for GLMs in [12]. However, the issue regarding estimation of standard error as discussed in the Lasso case is still carried over to the Group-Lasso as well.

Apart from the group structure, other types of structural associations between predictor variables have been modeled, like the fused Lasso [13] which imposes sparsity in the difference between successive coefficients, assuming a certain ordering of the variables. In this work, our focus is only on the grouped variable selection problem although extensions to other variations of structural associations can possibly be thought of along similar lines.

2.5 Flexibility in Inducing Sparsity

In high-dimensional data, an issue often associated with the Lasso formulation is the presence of too many non-zero coefficients in the solution. Using the Lasso, the usual way to remedy this problem is to shrink the coefficients further by increasing the c parameter in eq. (2.20). However, since the Lasso is based on global shrinkage of the coefficients, this results in shrinking even the non-zero coefficients, which in turn can effect the predictive accuracy of the estimated regression coefficients.

To address this issue, another version of the Lasso, namely the relaxed Lasso has been introduced in [14] which introduces an extra parameter ϕ in the following manner:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \|\boldsymbol{y} - X\{\boldsymbol{\beta}.\mathbf{1}_{M_c}\}\|_2^2 + \phi c\|\boldsymbol{\beta}\|_1, \tag{2.25}$$

where $c \geq 0$ and $\phi \in (0, 1]$ and $\mathbf{1}_{M_c}$ is an indicator function on the set of variables $M_c \subseteq \{1, ..., d\}$ so that the vector term $\{\boldsymbol{\beta}.\mathbf{1}_{M_c}\}$ has d components and for all $k \in \{1, ..., d\}$, each component is defined as:

$$\{\boldsymbol{\beta}.\mathbf{1}_{M_c}\}_k = \begin{cases} 0 & k \notin M_c \\ \beta_k & k \in M_c \end{cases}. \tag{2.26}$$

This formulation leads to a flexibility in imposing sparsity since the parameter c and ϕ separately control variable selection and shrinkage of coefficients.

The algorithm for relaxed Lasso breaks up the estimation into two steps, where the first step is the standard Lasso for producing the solution paths. The second step uses various sub-models along the path and again applies Lasso but with a small penalty parameter ϕc where $\phi \in [0, 1]$. As a result the relaxed Lasso finds the same set of sub-models as the Lasso but with less shrinkage of the non-zero coefficients. Another similar attempt towards a sparser solution with less shrinkage is the adaptive Lasso (see [15]).

Both approaches are designed so that the problem is still within the realm of convex optimization. An alternate approach can be to use the bridge regression penalization term

$\|\boldsymbol{\beta}\|_q$. For values of $q \leq 1$, the solutions produced would have the tendency to be sparse in nature below a certain threshold of κ. Figure 2.3 displays the constraint region for various values of q. To produce sparser solutions, one can optionally tune the parameter q along

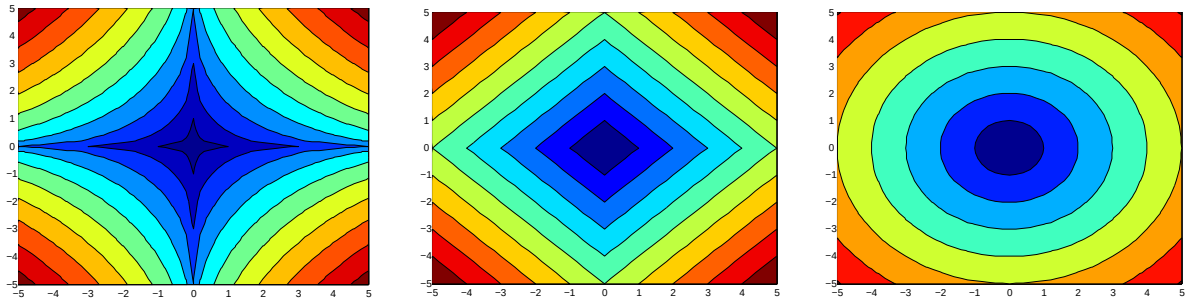


Figure 2.3.: This plot illustrates the difference in the feasible regions according to the different ℓ_q norms used. **Left**: Concentric circles with $\ell_{0.5}$ norm. **Center**: Concentric circles with ℓ_1 norm. **Right**: Concentric circles with ℓ_2 norm.

with κ, which gives an added flexibility and hence helps in avoiding excessive amounts of shrinkage of non-zero coefficients. Inspite of flexibility gains in the model, the overall optimization problem becomes non-convex and hence is harder to solve due to presence of local minima. We shall show in this work that the additional flexibility of adjusting sparsity without excessive shrinkage of regression coefficients can be easily achieved in a Bayesian framework by introducing an extra parameter similar to [14]. In the next section, we shift our focus to Bayesian inference and then move towards a Bayesian framework for variable selection.

2.6 Bayesian Inference

Although the Lasso based approach for variable selection has been very popular, there are still some shortcomings of this optimization based approach. The focus of an optimization based approach is to produce a single point estimate in the form of the MAP estimate. Although various estimates for the standard error have been suggested, they work under restricted conditions and in most cases only for the non-zero coefficients. Secondly, controlling the sparsity of the solution with a single parameter c (in eq. (2.20)) has a side-effect on the global shrinkage of the regression coefficients which may lead to decrease in prediction accuracy. Based on the issues associated with the optimization based framework for variable selection, there is a strong motivation to look at probabilistic approaches in order to overcome these issues. We will first discuss a general setting of the inference mechanism in a Bayesian setting and then discuss a Bayesian approach to single variable selection.

The advantage in using a Bayesian approach for modeling purposes is that it allows the extraction of information about the posterior distribution over the parameters in the model which usually includes estimating the moments and the mode of the distribution. Inference in a Bayesian setting generally refers to analyzing the posterior distribution. With

non-standard definition of priors and likelihoods, the posterior distribution is generally complex and as a result, quantities like the first and second moments cannot be derived analytically. In such cases, one has to resort to techniques which provide approximations to the desired estimates. One such popular inference mechanism that is used to generate such approximations is the Markov Chain Monte Carlo (MCMC) sampling technique. It involves generating a chain of sample points which under mild conditions, asymptotically converge to samples from the desired distribution. There are other forms of approximation in the Bayesian regime like Bayesian variational approximation. The choice for MCMC is driven by the fact that they are usually easy to setup and flexible with respect to the extensions in an existing framework. Next, we shall briefly review the basics of MCMC sampling and concepts related to Markov chains.

Markov Chain Monte Carlo (MCMC) Sampling. MCMC techniques are a class of methods based on random number generation which have been developed to deal with some of the issues that are encountered while working with complex probability distributions. Some these issues include finding expected values, normalization constants or analyzing other properties of a distribution over the parameters of a model ($\boldsymbol{\theta}$) which may require solving integrals. MCMC methods aim to find approximations to such values by generating parameter samples (where a "sample" is one instance of the parameter) which converge to samples from a target distribution, which in this case, is the distribution over $\boldsymbol{\theta}$.

A Markov chain ([16]) consists of a initial probability distribution ($p_{initial}$) which is used to sample the first point ($\boldsymbol{\theta}_0$) which we will call as the state of the chain at time $t = 0$. Subsequent data points are sampled using a transition probability distribution ($p_{transition}$) which defines a distribution for sampling a data point conditioned on the state of the chain in the previous time point. Hence starting with one data point ($\boldsymbol{\theta}_0$), subsequent samples are generated ($\boldsymbol{\theta}_1 \Rightarrow \boldsymbol{\theta}_2 \Rightarrow ..... \Rightarrow \boldsymbol{\theta}_n$). The Metropolis method is an example of a MCMC method and can be viewed as a generalization of another MCMC method called Gibbs sampling. We will discuss Gibbs sampling in more detail since that is extensively used in this work.

Consider a probability distribution p_{joint} over a d-dimensional vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_d\}$. We assume that it is not feasible to directly sample from this joint distribution, but it is possible to sample from the conditional distribution of one variable given all the others i.e $p(\theta_i|\boldsymbol{\theta}_{i-})$ is some standard distribution from which samples can be drawn easily. Here $\boldsymbol{\theta}_{i-}$ denotes a sub-vector which contain all the values of $\boldsymbol{\theta}$ except θ_i. In iteration ($t = 1$), We start by fixing the initial values of all variables to some random value $(\theta_1^1, \theta_2^1, ..., \theta_d^1)$ where the notation θ_i^j denotes the value of the i-th component in iteration j. In the next iteration ($t = 2$), each variable is sampled turn by turn fixing all the other variables to their most recently sampled value. The sampling at the $(t+1)$-th iteration is as follows:

$$\begin{aligned} \theta_1^{t+1} &\sim p(\theta_1^{t+1}|\boldsymbol{\theta}_{1-}^t) \\ \theta_2^{t+1} &\sim p(\theta_2^{t+1}|\boldsymbol{\theta}_{2-}^t) \\ \theta_d^{t+1} &\sim p(\theta_d^{t+1}|\boldsymbol{\theta}_{d-}^t). \end{aligned} \tag{2.27}$$

Each iteration results in a sample vector $\boldsymbol{\theta}^{t+1} = (\theta_1^{t+1}, \theta_2^{t+1}, ..., \theta_d^{t+1})$. The samples are generated using a first order Markov chain i.e. each point is sampled conditioned only on the sample point generated in the previous step in the chain. Under mild conditions, asymptotically, as $t \to \infty$, the samples collected in this manner $(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3, ..., \boldsymbol{\theta}^t)$ converge to the samples from p_{joint}. Using these samples, it is possible to approximate certain quantities like for example, the expected value of the distribution can be estimated by averaging the samples i.e. $(\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{\theta}^t)$.

A variation of the Gibbs sampler is the blocked-Gibbs sampler, in which the conditional sampling step combines a few variables, referred to as a block and these variables are then sampled from their joint distribution in one step rather than individually for faster convergence of the Markov chain.

Convergence of a Markov Chain. Before commenting on the convergence rate of an MCMC sampler, we define some terms related to Markov chains which will help explain the notion of convergence.

In a Markov chain $(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3, ..., \boldsymbol{\theta}^t)$, the state space is defined as all the possible values of $\boldsymbol{\theta}^i$. For continuous state spaces, this is generally $\mathbb{R}^d$. A state $\boldsymbol{\theta}^c$ has period k if any return to state $\boldsymbol{\theta}^c$ must occur in multiples of k time steps. If $k = 1$, then the state is said to be aperiodic i.e. returns to state $\boldsymbol{\theta}^c$ can occur at irregularly. A state $\boldsymbol{\theta}^c$ is said to be transient if, given that we start in state $\boldsymbol{\theta}^c$, there is a non-zero probability that we will never return back to the same state. A state $\boldsymbol{\theta}^c$ is recurrent if it is not transient. Positive recurrence implies that the expected time for recurrence to occur is finite.

A Markov chain is said to be ergodic if all states i.e. all possible values of $\boldsymbol{\theta}^i$, are aperiodic and positive recurrent. Ergodicity primarily establishes the convergence of a Markov chain to its stationary distribution where $\pi(\boldsymbol{\theta})$ is a stationary probability distribution of Markov chain if:

$$\int_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta}) p(\boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{\theta} = \pi(\boldsymbol{z}), \tag{2.28}$$

where $p(\boldsymbol{z}|\boldsymbol{\theta})$ is the transition probability distribution. In the context of MCMC methods, the stationary distribution is the desired target distribution from which samples need to be generated.

A useful property of the Markov chains especially with respect to sampling from a target distribution is the rate of convergence. The rate of convergence is measured by how soon the samples (in terms of length of the chain) get closer to samples from the target distribution. Faster convergence means that the samples are nearer to the target distribution with less number of samples collected. Fast convergence is indicated by a property called as geometric ergodicity. A chain is said to be geometrically ergodic if there is a constant $0 \leq \tau < 1$ and a real integrable function $M(\boldsymbol{\theta})$ such that:

$$\|p^n(\boldsymbol{\theta}, \cdot) - \pi(\cdot)\| \leq M(\boldsymbol{\theta})\tau^n, \tag{2.29}$$

where p denotes the transition probability, n denotes the number of iterations and π denotes the stationary distribution.

After describing the basic ideas related to Bayesian inference, we now look at a probabilistic model of variable selection with the inference specified in terms of MCMC sampling.

2.7 Bayesian Variable Selection

Along with the optimization based formulation of the Lasso, a probabilistic interpretation was also mentioned in [4], where the prior over regression coefficients was defined as the product of independent double exponential or Laplacian distributions. However, a detailed inference mechanism was missing. A fully Bayesian model of the Lasso, called Bayesian Lasso, along with an efficient inference mechanism in the form of MCMC sampling is formulated in [17].

The probabilistic model for the Bayesian Lasso (as in [17]) with normally distributed likelihood is defined as follows:

$$\begin{aligned} \boldsymbol{y} &\sim N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2 I) \\ \boldsymbol{\beta} &\sim \prod_{i=1}^{d} \text{Lap}(\beta_i|0, \sqrt{\rho}/\sigma), \end{aligned} \tag{2.30}$$

where "Lap" denotes the Laplacian distribution defined as:

$$\text{Lap}(a|0, c/d) \propto \frac{c}{2d} \exp(-c|a|/d). \tag{2.31}$$

This model is directly derived from the Lasso formulation by rewriting it in a probabilistic manner. The posterior distribution over the regression coefficients is as follows:

$$p(\boldsymbol{\beta}|\boldsymbol{y}, X, \sigma^2, c) \propto N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2 I) \prod_{i=1}^{d} 0.5\sqrt{\frac{\rho}{\sigma^2}} \exp(-\sqrt{\rho}|\beta_i|/\sigma). \tag{2.32}$$

The MAP solution to above model is equivalent to the Lasso solution which can be seen by taking negative log likelihood of eq. (2.32) which gives us:

$$\boldsymbol{\beta}_{MAP} = \text{argmin}_{\boldsymbol{\beta}}(\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \sqrt{\rho\sigma^2}\|\boldsymbol{\beta}\|_1), \tag{2.33}$$

which is equivalent to eq. (2.20). The Lagrangian parameter in this case is represented by (σ^2, ρ) and can be viewed as model selection parameters. Looking at eq. (2.32), we see that the posterior is a complex distribution and it is difficult to analytically derive quantities like first and second moments. Alternatively, the Laplacian prior can be rewritten as a

scale-mixture of normals [18]:

$$\begin{aligned} p(\boldsymbol{\beta}|\rho,\sigma^2) &\propto \prod_{i=1}^{d} \frac{\sqrt{\rho}}{2\sigma} \exp(-\sqrt{\rho}|\beta_i|/\sigma) \\ &\propto \prod_{i=1}^{d} \int_0^\infty N(\beta_i|0,\sigma^2\lambda_i^2)\text{Expd}\left(\lambda_i^2|1,\frac{\rho}{2}\right) d\lambda_i^2, \end{aligned} \tag{2.34}$$

where "Expd" denotes the exponential distribution (see appendix A). The integral can be solved analytically and results in the Laplacian prior defined in eq. (2.30). Further, for a fully Bayesian model, a prior is defined on σ^2. This results in the following hierarchical model which introduces the auxiliary variables Λ:

$$\begin{aligned} \boldsymbol{y} \sim& N(\boldsymbol{y}|X\boldsymbol{\beta},\sigma^2 I) \\ \boldsymbol{\beta} \sim& \prod_{i=1}^{d} N(\beta_i|0,\sigma^2\lambda_i^2) \\ \Lambda \sim& \prod_{i=1}^{d} \text{Expd}\left(\lambda_i^2|1,\frac{\rho}{2}\right) \\ \sigma^2 \sim& \text{Inv-}\chi^2(\sigma^2|\nu_0,s_0), \end{aligned} \tag{2.35}$$

where Λ is a diagonal matrix with diagonal entries $(\lambda_1,\lambda_2,...,\lambda_d)$ and Inv-χ^2 denotes the inverse chi-square distribution. The joint posterior over the variables is written as:

$$\begin{aligned} p(\boldsymbol{\beta},\Lambda|\boldsymbol{y},X,\sigma^2,c) \propto& N(\boldsymbol{y}|X\boldsymbol{\beta},\sigma^2 I)\prod_{i=1}^{d} N(\beta_i|0,\sigma^2\lambda_i^2) \\ &\cdot\left[\prod_{i=1}^{d}\text{Expd}\left(\lambda_i^2|1,\frac{\rho}{2}\right)\right]\text{Inv-}\chi^2(\sigma^2|\nu_0,s_0), \end{aligned} \tag{2.36}$$

The model is graphically illustrated in Figure 2.4. This hierarchical model makes posterior inference feasible via MCMC sampling. In particular, since all the conditional distributions are of standard form, Gibbs sampling can be applied easily. The auxiliary variables and the model selection parameters are integrated out stochastically during sampling. As a result, samples thus obtained can be used to summarize the posterior distribution over the regression coefficients with estimates for the expectation and variance. Also, as shown in [17], the sharing of a unique σ^2 parameter for the whole model guarantees a unimodal joint posterior distribution over $(\boldsymbol{\beta},\sigma^2)$ for typical choices of marginal prior over σ^2, which helps in avoiding a slower convergence of the Gibbs sampler. It has been shown in [8] that the Gibbs sampler for Bayesian Lasso hierarchical model is geometrically ergodic which indicates rapid convergence of the sampler.

The expectation of the posterior distribution over the regression coefficients is not sparse. Hence, unlike the classical Lasso, variable selection cannot be done trivially by inclusion

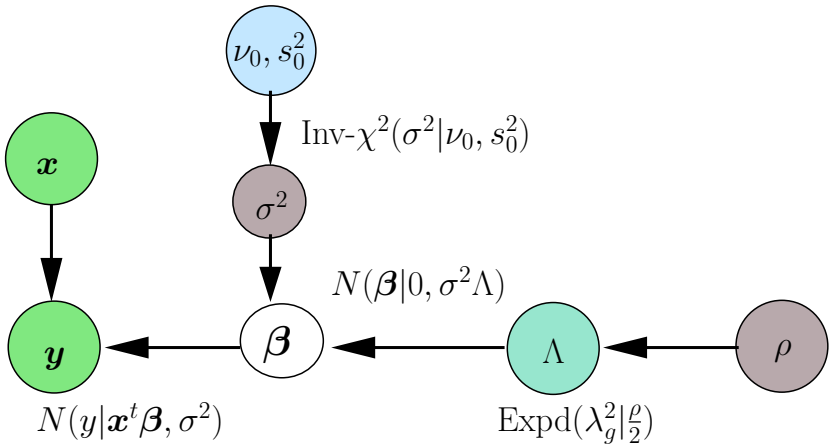


Figure 2.4.: The full hierarchical model for the Bayesian Lasso along with the auxiliary variables Λ. The green circles indicate data related variables. The brownish circles denote the Lagrangian parameters.

of non-zero coefficient values from an estimate of the expectation. To further produce a sparse point estimate, heuristics such as thresholding have been used for variable selection (see [19],[20]). We show in later chapters that there is a more principled way within the Bayesian framework to obtain a sparse coefficient vector which automates variable selection without the need for an extra thresholding step.

As in the classical Lasso, the Bayesian model also depends on a fixed set of parameters (σ^2, ρ) which jointly control sparsity and global shrinkage of coefficients. Hence sparser solutions are associated with the global shrinkage of the non-zero coefficients which may affect the predictive power of the inferred model. Hence it is beneficial to have an additional parameter which can control sparsity of the solution with a reduced coupling with the global shrinkage of regression coefficients.

Additional Parameter for Flexibility in Sparsity. Based on the problem discussed above regarding lack of flexibility in tuning sparsity levels, a further extension to the Bayesian Lasso involves introducing an additional parameter. This parameter provides flexibility to the model in terms of specifying the level of desired sparsity in the solution without compromising too much on the global shrinkage of the regression coefficients. Such a flexibility is achieved in [21] by introducing a shape parameter in the distribution over the auxiliary variables Λ. From an optimization point of view, this can be interpreted as adding a parameter which alters the shape of the feasible region to tune the level of sparsity in the solution. The effect is similar to adding l_p norms constraints to a least squares problem where $p \leq 1$. The modified Bayesian sparse variable selection model in [21] changes the distribution over the auxiliary variables λ_i's to a gamma distribution by

adding a new parameter α:

$$\Lambda \sim \prod_{i=1}^{d} \text{Gamma}(\lambda_i|\alpha, \rho/2), \tag{2.37}$$

where "Gamma" denotes the gamma distribution. We obtain the Bayesian Lasso as a special case by setting $\alpha = 1$. For $\alpha \leq 1$, this model is sparsity inducing. The modified hierarchical model is shown in Figure 2.5.

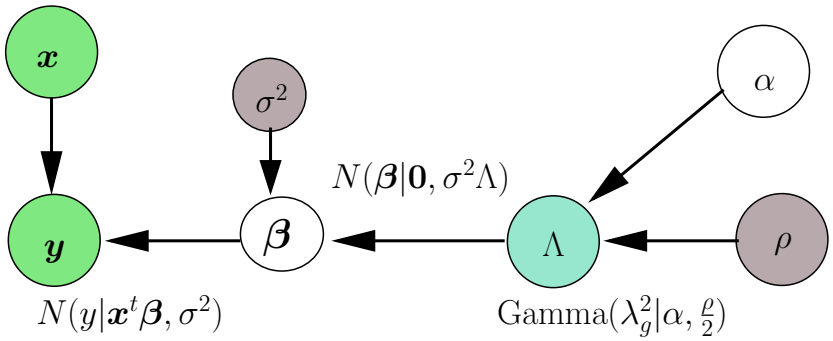


Figure 2.5.: The full hierarchical model for the Bayesian Lasso with the auxiliary variables defined for inference purposes. The green circles indicate data related variables. The brown circles denote the Lagrangian parameters.

2.8 Summary

In this chapter, we reviewed some basic concepts related to regression in linear models. In the context of linear regression models, some existing methods for variable selection were discussed. We started first with the problem formulation for single variable selection in an optimization framework, focusing mainly on the Lasso since it motivates the work in this thesis. We then described the Group-Lasso, which is a generalization of the Lasso where selection is carried out in groups of predictor variables. A common problem with these methods are the lack of meaningful variance estimates.

We then turned our attention to Bayesian inference and reviewed some of concepts and methods that will be used in the subsequent chapters. Using these concepts, we described an existing Bayesian version of the Lasso for single variable selection which addresses the problem of variance estimation and hence motivates our work with regard to a generalization of the Bayesian Lasso to Bayesian grouped-variable selection. Further, we also saw the need for additional flexibility with regard to specifying the level of sparsity, which is achieved in the context of single variable selection by adding an extra parameter for tuning the level of sparsity in the solution. In the next chapter, we will lay the foundations of Bayesian grouped variable selection by using some of ideas discussed in this chapter.

3

Grouped-Variable Selection in Linear Regression Models

3.1 Towards Bayesian Grouped Variable Selection

In the previous chapter, we described some of the existing ideas regarding Bayesian variable selection. Motivated by these ideas, we list out the tasks to be accomplished as a part of this thesis. These tasks will be described and executed in detail subsequently. Our broad focus in this work is to extend and generalize the idea of Bayesian variable selection and encapsulate these extensions in a unified inference framework:

- Our first task will be to extend the Bayesian Lasso for grouped-variable selection. We shall call this extension the Bayesian Group-Lasso.
- Next, we will add further flexibility to the Bayesian Group-Lasso in controlling the level of sparsity. This will be achieved by adding another parameter which helps in tuning the level of sparsity in the solution with a milder effect on the global shrinkage of regression coefficients.
- We will also provide a more principled approach to variable selection as opposed to heuristics like thresholding.
- We will expand the usage of this framework to various generalized linear models in order to cater to a wide variety of applications.
- Finally, in the context of survival analysis, we will look at the extension of grouped-variable selection to a clustering model where variable selection for each cluster will be done simultaneously with cluster identification.

3.2 Grouped-Variable Selection

In the previous chapter, we discussed the problem of individual variable selection in the context of linear regression. We also looked at some of the methods that exist in literature to address this problem both from an optimization and a Bayesian perspective. We observed that the Bayesian view of the problem helped in extracting different estimates

from the posterior distribution over the regression coefficients which was missing in the optimization view of the problem. However, in various application settings, the variables are endowed with a natural group structure. This, in turn, leads to a changed notion of variable selection, wherein it is more desirable to select whole groups of related variables rather than individual variables. Some common examples of such problems include the k-th order polynomial expansions of the input variables where the groups consist of products over combination of variables. Another popular example is the data consisting of categorical variables (i.e. "factors" in the usual statistical terminology) and optionally, their interactions. Such variables are then represented as **groups** of dummy variables. In such application settings, it makes more sense to select entire groups of dummy variables which represent a single categorical variable. This requirement leads to the need for formulating a grouped variable selection problem. Such type of categorical data is commonly encountered especially in biological applications. Examples of such applications include bio-marker identification, birth weight prediction, splice site detection etc.

In the context of penalized linear regression, this modified problem of grouped variable selection has been addressed in the optimization based framework via the **Group-Lasso** as defined in [11]. This formulation is motivated by the classical Lasso, where the ℓ_1 norm constraint is replaced with a ℓ_1-ℓ_2 norm constraint, where the ℓ_1 part of it is applied between groups to encourage sparsity in groups of variables. This formulation has been applied to a variety of applications and its properties have been analyzed theoretically (see [12], [22] and [23]). As with the classical Lasso, this optimization based view of the Group-Lasso provides a MAP estimate but lacks in providing other estimates of the posterior distribution of the regression coefficients like the expectation and variance estimates. Similar to the Bayesian Lasso [17], it would be beneficial to have a probabilistic formulation of the grouped variable selection problem to estimate various quantities from the posterior distribution of the regression coefficients. Hence with a motivation similar to the Bayesian Lasso, we introduce a framework for **Bayesian Group-Lasso** in order to perform grouped-variable selection and to estimate various quantities like the first and second moments which would provide more information about uncertainty in the variable selection process.

Apart from the goal of providing a probabilistic framework for grouped-variable selection in order to summarize the posterior distribution over regression coefficients, an additional goal is to introduce flexibility in the level of sparsity imposed on the solution while avoiding excess global shrinkage of the coefficients. This is motivated by the observation that the Lasso has only a single parameter to control sparsity which simultaneously also affects the global shrinkage [14]. Traditionally, sparsity of the solution has been tuned by adjusting the regularization parameter which reduces the number of selected variables but as a consequence results in the global shrinkage of the non-zero regression coefficients. This in turn may result in a poorer predictive performance. To overcome this limitation of the model, a generalization for Bayesian Lasso has been introduced in [21] where an extra parameter has been added to create a flexible class of priors which can enforce varying levels of sparsity. On similar lines, we incorporate this extension for our Bayesian grouped-variable selection framework.

In the following sections, we again describe the Group-Lasso optimization problem.

We then define our Bayesian interpretation of the same problem, with an extension to flexible sparsity. This is followed by the description of the inference algorithm and finally concluded with experiments to demonstrate the workings of the inference mechanism.

3.3 Group-Lasso

We first describe the classical Group-Lasso formulation. In this modified problem of grouped selection, as before, we have n responses $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ and an $n \times d$ design matrix X which consists of n observation vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ arranged as rows in X which we represent as $(\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, ..., \boldsymbol{x}_n^t)$. We can also view the matrix X in terms of its d columns consisting of the column vectors $(\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_d)$ where each column represents a predictor variable. Additionally, the predictor variables are grouped into G groups with size of each group denoted by p_g. Hence the design matrix can be represented as sub-matrices which include column vectors of a group of predictor variables i.e $X = \{X_1, X_2, ..., X_G\}$ where each sub-matrix $X_g = \{\boldsymbol{c}_{g_1}, \boldsymbol{c}_{g_2}, ..., \boldsymbol{c}_{g_{p_g}}\}$ represents a single group of columns or predictor variables and g_i is the index of a column in that group. Similarly, the regression coefficients $\boldsymbol{\beta}$ can also be split into groups of coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_G)$ where the components of $\boldsymbol{\beta}_g$ have a one-to-one correspondence to the columns of X_g. The breakup of groups is graphically shown in Figure 3.1.

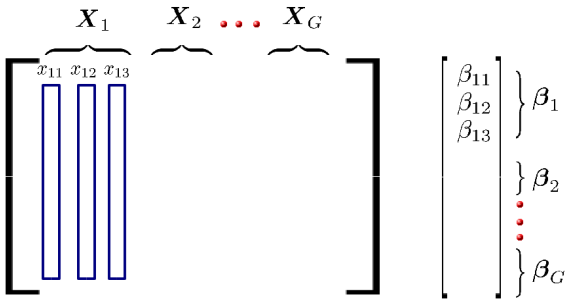


Figure 3.1.: Design matrix X is divided into sub-matrices $\{X_1, X_2, ..., X_G\}$ where each sub-matrix represents a group. Similarly the regression coefficients $\boldsymbol{\beta}$ are also divided into groups with a one-to-one correspondence with the sub-matrices.

The goal of inference is to select significant groups as a whole rather than individual variables. The modified constrained regression problem, termed as the Group-Lasso, is stated as follows:

$$\text{argmin}_\beta \, \mathcal{C}(\boldsymbol{\beta}) \quad \text{s.t.} \quad \sum_{g=1}^{G} \|\boldsymbol{\beta}_g\|_2 \leq \kappa, \tag{3.1}$$

where $\mathcal{C}(\boldsymbol{\beta})$ is the cost function. This definition is modified further to rescale the constraint based on the size of each group:

$$\text{argmin}_\beta \, \mathcal{C}(\boldsymbol{\beta}) \quad \text{s.t.} \quad \sum_{g=1}^{G} \sqrt{p_g} \|\boldsymbol{\beta}_g\|_2 \leq \kappa \tag{3.2}$$

As mentioned in Chapter 2, we will specifically deal with likelihoods which stem from the

exponential family of distributions which implies that $\mathcal{C}(\boldsymbol{\beta})$ will be a convex function for the models that we will consider. For least squares regression, the likelihood is based on the normal distribution and hence $\mathcal{C}(\boldsymbol{\beta}) = \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2$. Based on eq. (3.2), the goal is then to minimize $\mathcal{C}(\boldsymbol{\beta})$ under a constraint on the sum of the ℓ_2-norms of the sub-vectors $\boldsymbol{\beta}_g$, where $\ell_p(\boldsymbol{m}) = (\sum_{i=1}^{d} |m_i|^p)^{1/p}$ for a d-dimensional vector $\boldsymbol{m}$. As described in the previous chapter, for $p \geq 1$, ℓ_p norm is convex whereas for $0 < p < 1$, l_p defines a semi-norm and hence is non-convex. Hence for Group-Lasso, since the constraint consists of a sum of ℓ_2 norms, the feasible region is convex and hence the overall problem is a convex optimization problem which can be solved efficiently (see [12], [11]). It also immediately follows from eq. (3.2) that the Lasso is a special case of Group-Lasso where the group size $p_g = 1$ for all groups.

In the next section, we interpret the Group-Lasso in a Bayesian framework by defining a suitable prior motivated by the prior specification in the Bayesian Lasso.

3.4 The Bayesian Group-Lasso

Similar to the Bayesian Lasso, we begin by reformulating the Group-Lasso for a standard linear regression model with normally distributed noise in a probabilistic framework which we will call the Bayesian Group-Lasso. We use a normally distributed likelihood and a product of multivariate Laplacian priors over the regression coefficients. We define the probabilistic model in the following manner:

$$\begin{aligned} \boldsymbol{y} &\sim N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2 I) \\ \boldsymbol{\beta}_g &\sim \text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, c^{-1}) \quad \forall g = 1 \dots G, \end{aligned} \tag{3.3}$$

where M-Laplace is a spherical p_g dimensional multivariate Laplacian distribution defined over each group of regression coefficients as follows:

$$\text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, c^{-1}) \propto c^{p_g/2} \exp(-c\|\boldsymbol{\beta}_g\|_2). \tag{3.4}$$

Hence, given n observations, the posterior distribution of the regression coefficients can be written as:

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{y}, X, \sigma^2, c) \propto & N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2) \prod_{g=1}^{G} \text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, c^{-1}) \\ \propto & N(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2) \prod_{g=1}^{G} \text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, c^{-1}). \end{aligned} \tag{3.5}$$

Taking negative log likelihood of eq. (3.5) gives us back the Lagrangian form of the Group-Lasso formulation in eq. (3.2) and hence finding a MAP solution to eq. (3.5) becomes equivalent to optimizing the Group-Lasso described in eq. (3.2), with c playing the role of a fixed Lagrange parameter. Our primary goal is to develop an inference algorithm which

can enable us to summarize the posterior distribution over regression coefficients with the moments and mode of the distribution. Since it is not feasible to analytically derive all these quantities from the above formulation, it is necessary to find an alternate formulation of eq. (3.5) in order to make posterior analysis feasible. Our focus is to reformulate the model in a way such that MCMC sampling can be used for inference purposes in the same manner as was done in the Bayesian Lasso. We now discuss an alternate representation of the prior over regression coefficients which will be used to build our framework for inference.

3.4.1. Prior Formulation

To make posterior analysis feasible, we reformulate the prior via the introduction of latent variables using a hierarchical model. A latent variable representation of the Laplacian distribution is the scale-mixture of normals ([17],[18]). We extend the same concept to rewrite the multivariate Laplacian distribution over the regression coefficients as follows:

$$\begin{aligned}\prod_{g=1}^{G} p(\boldsymbol{\beta}_g|\bullet) &= \prod_{g=1}^{G} \text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, c^{-1}) \\ &= \prod_{g=1}^{G} \int_0^\infty N(\boldsymbol{\beta}_g|\mathbf{0}, \sigma^2\lambda_g^2 I)\text{Gamma}(\lambda_g^2|\tfrac{p_g+1}{2}, \tfrac{a_g}{2})\, d\lambda_g^2,\end{aligned} \tag{3.6}$$

where $a_g = p_g\rho$ and λ_g's are the auxiliary variables. It can be shown analytically that this scale-mixture of normals is equivalent to the multivariate Laplacian distribution defined earlier:

$$\begin{aligned}p(\boldsymbol{\beta}_g|\rho, \sigma^2) &= \int_0^\infty N(\boldsymbol{\beta}_g|\mathbf{0}, \sigma^2\lambda_g^2 I)\text{Gamma}(\lambda_g^2|\tfrac{p_g+1}{2}, \tfrac{a_g}{2})\, d\lambda_g^2 \\ &= (\sigma^2)^{-\frac{p_g}{2}} \int_0^\infty \underbrace{(\lambda_g^2)^{-\frac{1}{2}} \exp\left[-\frac{b_g}{2\lambda_g^2} - \lambda_g^2\frac{a_g}{2}\right]}_{(\frac{b_g}{a_g})^{\frac{1}{4}} K_{\frac{1}{2}}\left[(a_g b_g)^{\frac{1}{2}}\right]\cdot \text{GIG}(\lambda_g^2|\frac{1}{2}, a_g, b_g)} a_g^{\frac{p_g+1}{2}} d\lambda_g^2 \\ &\propto (\sigma^2)^{-\frac{p_g}{2}} (\tfrac{b_g}{a_g})^{\frac{1}{4}} (a_g b_g)^{-\frac{1}{4}} a_g^{\frac{p_g+1}{2}} \exp\left[-(a_g b_g)^{\frac{1}{2}}\right] \\ &\propto (a_g/\sigma^2)^{\frac{p_g}{2}} \exp(-(a_g/\sigma^2)^{\frac{1}{2}}\|\boldsymbol{\beta}_g\|_2) \\ &\propto \text{M-Laplace}(\boldsymbol{\beta}_g|\mathbf{0}, (a_g/\sigma^2)^{-\frac{1}{2}}),\end{aligned} \tag{3.7}$$

where GIG is the generalized inverse Gaussian distribution defined in Appendix A, $b_g = \frac{\|\boldsymbol{\beta}_g\|^2}{\sigma^2}$ and $K_v(\cdot)$ is the modified Bessel function of the second kind and $c = \sqrt{\frac{p_g\rho}{\sigma^2}}$. It is important to note here that this particular formulation of the prior is not the only way to specify sparsity inducing distributions as a scale-mixture of normal distributions. Other formulations have been used recently like the one defined in [24], which uses a single multivariate Laplacian distribution over all the regression coefficients.

The parameters ρ and σ play the role of the Lagrangian parameter $\left(= \sqrt{\rho\sigma^2}\right)$ in the

penalized version of the Group-Lasso defined in eq. (3.2). Hence (ρ, σ^2) can be viewed as model selection parameters. These parameters can be determined using cross-validation. We opt for a more convenient alternative of specifying a prior over these parameters and integrating them out stochastically.

3.4.2. Hyperpriors

In this section, we define hyperpriors over the model selection parameters σ^2 and ρ for a full Bayesian treatment of the model, as opposed to learning them by cross-validation. The key motivation behind breaking up the prior on $\boldsymbol{\beta}$ into a hierarchical model, by introducing latent variables Λ, was to make posterior analysis feasible. The inference mechanism that we choose later is primarily driven by the ability to sample from the posterior conditional distributions of all the variables involved. Hence the choice of priors on variables σ^2 and ρ is also driven by the simplification of the inference process, which in turn means being able to sample from the posterior conditional distributions of these variables.

For σ^2, we define a standard conjugate prior (see [25]) as:

$$p(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{\beta}|\sigma^2)p(\sigma^2) = N(\boldsymbol{\beta}|\mathbf{0}, \sigma^2\Lambda) \cdot \text{Inv-}\chi^2(\sigma^2|\nu_0, s_0^2), \tag{3.8}$$

where Λ is a $d \times d$ diagonal matrix with diagonal entries from λ_1 to λ_G, with each λ_g repeated p_g times. The Λ matrix is illustrated in Figure 3.2. For ρ, a conjugate gamma

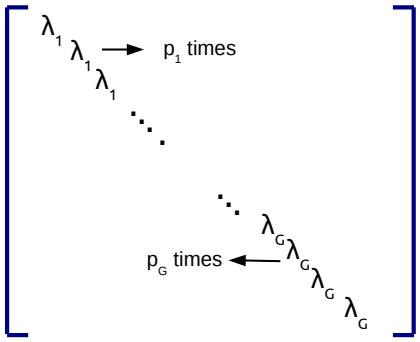


Figure 3.2.: The Λ matrix is constructed as a diagonal matrix with diagonal entries from λ_1 to λ_G, with each λ_g repeated p_g times.

prior is defined as follows:

$$p(\rho|r, s) \propto \rho^{Gr-G} \exp\left(-\frac{\rho G}{s}\right). \tag{3.9}$$

With these hyperpriors for ρ and σ^2, all the posterior conditionals are now of standard form. Instead of choosing a particular value for these parameters (as in cross-validation), these variables will be integrated out through the inference procedure.

In certain experiments with Bayesian variable selection, such a procedure has shown to be more effective for predictive performance than the traditional cross-validation step for choosing model selection parameters (see [26]). With a particular model of Bayesian Lasso regression, a comparison experiment was conducted in [26] with the diabetes data in [7] to compare the prediction errors of the standard Lasso regression to a Bayesian version of Lasso regression. In the standard Lasso regression, model selection was done via cross-validation, whereas the Bayesian setting integrated out these parameters through MCMC sampling. The results showed that 65.6% of the time the Bayes procedure performed better which indicates the possibility of predictive gains while using the fully Bayesian model. The full hierarchical model for Bayesian Group-Lasso in terms of the latent variables is shown in Figure 3.3.

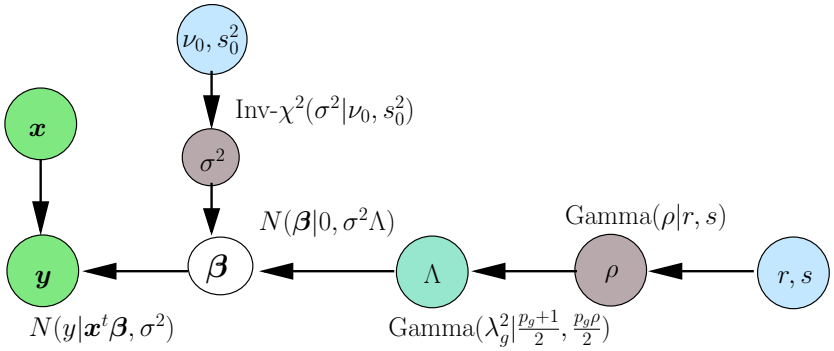


Figure 3.3.: The full hierarchical model for the Bayesian Group-Lasso - The different colors indicate different roles of the variables in the model. Green - given observations, cyan - auxiliary variable, brownish - model selection parameters, blue - fixed hyperparameters.

Setting the Hyperparameter Values. We briefly discuss the setting of hyperparameter values in the model. There are four parameters, ν_0, s_0^2, r and s which need to be fixed. The first two parameters define the Inv-χ^2 distribution and hence can be interpreted as ν_0 being the number of additional virtual samples available with a variance of s_0^2. The only other hyperparameters in the model are the shape r and scale s in the Gamma(r, s) prior on ρ. Testing for suitable values can be done straightforwardly by first sampling ρ from the hyperprior and in turn sampling λ_g^2 from Gamma$(\lambda_g^2|\frac{p_g+1}{2}, \frac{p_g\rho}{2})$. The fraction of "large" λ_g^2 values encodes our prior belief about the sparsity of the model. Since λ_g^2 is a variance parameter for the g-th group of regression coefficients, the fraction of large λ_g^2 values essentially determines the expected sparsity.

3.4.3. Generalized Sparsity

Although the hierarchical prior that we have defined so far is a sparsity inducing distribution, it is still restrictive since there is only one way of tuning the sparsity of the solution.

This is done by tuning the parameters (σ^2, ρ) with the intention of suppressing some more regression coefficients towards zero. However, this results in a global effect of further shrinking all the regression coefficients including the significant ones. Although, it serves the purpose of variable selection, this excessive shrinking of the regression coefficients may adversely affect the predictive performance of the model, as shown in the experiment in section 6.6.2.

As described in the previous chapter, the work in [21] shows how an extra parameter can be added to the Bayesian single variable selection model to provide more flexibility in generating sparser solutions. Following the work in [21], we introduce an extra parameter α in the model in order to produce prior distributions over $\boldsymbol{\beta}$ which are capable of varying the sparsity of the solution without excessively shrinking the significant regression coefficients. The α parameter is introduced as a part of the shape parameter in the gamma distribution of λ_g^2 (defined in eq. 3.6) as follows:

$$\prod_{g=1}^{G} p(\boldsymbol{\beta}_g) = \prod_{g=1}^{G} \int_0^\infty N(\boldsymbol{\beta}_g|\mathbf{0}, \sigma^2\lambda_g^2 I)\text{Gamma}(\lambda_g^2|\alpha\tfrac{p_g+1}{2}, \tfrac{a_g}{2})\, d\lambda_g^2. \tag{3.10}$$

Based on this changed hierarchical prior, we can derive the marginal probability density function for $\boldsymbol{\beta}_g$, by using the definition of a generalized inverse Gaussian distribution [27] (see Appendix A):

$$\begin{aligned} p(\boldsymbol{\beta}_g|\sigma) &= \textstyle\int_0^\infty N(\boldsymbol{\beta}_g; \mathbf{0}, \sigma^2\lambda_g^2 I)\ p(\lambda_g^2)\ d\lambda_g^2 \\ &= \frac{(\sigma^2)^{-\frac{p_g}{2}}}{\sqrt{2\pi}\ \Gamma(p_g'\alpha)} \int_0^\infty (\lambda_g^2)^{p_g'\alpha-\frac{p_g}{2}-1}\ \exp\left(-\frac{1}{2}\left[\frac{b_g}{\lambda_g^2} + \lambda_g^2 p_g\rho\right]\right)\left(\frac{p_g\rho}{2}\right)^{p_g'\alpha} d\lambda_g^2 \\ &= \frac{(\sigma^2)^{-\frac{p_g}{2}}\ b_g^{\frac{1}{2}(p_g'\alpha-\frac{p_g}{2})}\ K_{(p_g'\alpha-\frac{p_g}{2})}(\sqrt{p_g\rho b_g})\ \ (p_g\rho)^{(p_g'\alpha+\frac{p_g}{2})}}{\sqrt{\pi}\ \ \Gamma(p_g'\alpha)\ \ 2^{(p_g'\alpha-\frac{1}{2})}}, \end{aligned} \tag{3.11}$$

where $p_g^{'} = \frac{p_g+1}{2}$, $b_g = \frac{\|\beta_g\|_2^2}{\sigma^2}$ and $K_\nu(.)$ is the modified Bessel function of the second kind.

The sparsity inducing nature of the prior is determined by the value of α. For values $\alpha \leq 1$, this prior becomes a sparsity inducing distribution, where $\alpha = 1$ corresponds to the Bayesian Group-Lasso case. For $\alpha > 1$, the prior is no longer sparsity inducing and closely resembles a normal distribution for $\alpha = 2$. Figure 3.4 shows the 2-D contour plots for various types of prior distributions induced by setting different α values.

Prior on α. Since we have introduced another parameter α in the model, we can define a prior on this variable as well. Based on the work in [28], we define a joint conjugate prior on (ρ, α):

$$p(\alpha, \rho|t, q, r, s) \propto \frac{t^{\alpha-1}\exp(-\rho q)}{\Gamma(\alpha)^r \rho^{-\alpha s}}. \tag{3.12}$$

The modified hierarchical model is shown in Figure 3.5.

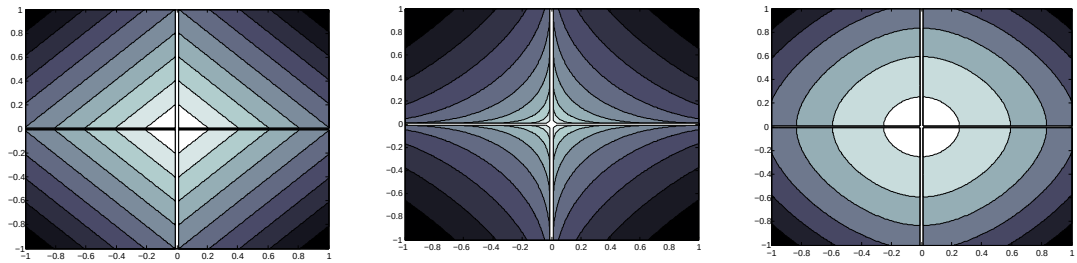


Figure 3.4.: 2-D Plots of the prior over $\boldsymbol{\beta}$ for different values of α. **Left:** Plot of the prior for $\boldsymbol{\beta}$ for $\alpha = 1$ which resembles the Lasso constraint. **Center:** Plot of the prior for $\boldsymbol{\beta}$ for $\alpha = 0.5$. **Right:** Plot of the prior for $\boldsymbol{\beta}$ for $\alpha = 2.0$ which resembles a normal distribution.

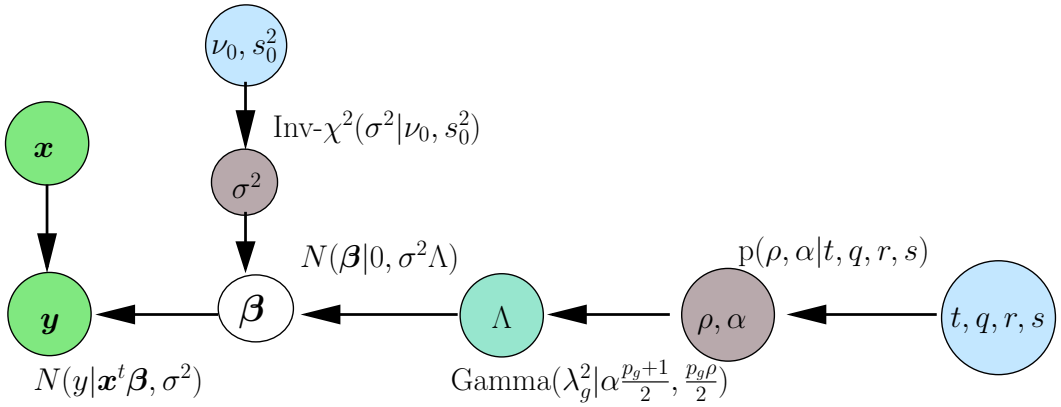


Figure 3.5.: The full hierarchical model for Bayesian grouped-variable selection with an extra parameter α for controlling the level of sparsity. The prior distribution over (ρ, α) is based on eq. 3.12.

3.5 Posterior Inference via MCMC Sampling

As mentioned before, it is not feasible to analyze the posterior distribution over regression coefficients. The primary motivation behind creating a hierarchical prior over the regression coefficients was to aid inference. Since the re-construction of the prior resulted in all the posterior conditional distributions to be of standard form, a natural choice for inference is Gibbs sampling which is a type of Markov Chain Monte Carlo sampling technique. As described in the previous chapter, the method entails sampling from a joint distribution of variables by sampling from the conditional distribution of each parameter in turn, keeping others fixed. Under mild conditions, asymptotically, samples thus generated converge to the samples from the true joint distribution of the variables. To derive all the posterior conditional distributions of our model, we consider the joint posterior distribution for all

the parameters:

$$\begin{aligned} p(\boldsymbol{y}, X, \boldsymbol{\beta}, \Lambda, \sigma^2, \rho) \propto (\sigma^2)^{-\frac{d}{2}} \exp\left[-\frac{1}{2}\cdot\sigma^{-2}(\boldsymbol{y}-X\boldsymbol{\beta})^t(\boldsymbol{y}-X\boldsymbol{\beta})\right] \\ \cdot (\sigma^2)^{-\frac{d}{2}} \prod_{g=1}^{G}\left[(\lambda_g^2)^{-\frac{p_g}{2}} \exp\left(-\frac{\|\boldsymbol{\beta}_g\|^2}{2\lambda_g^2\sigma^2}\right)\right. \\ \left.\cdot (\lambda_g^2)^{\frac{(p_g+1)}{2}-1}\rho^{\frac{(p_g+1)}{2}} \exp\left(-\lambda_g^2\frac{p_g\rho}{2}\right)\right] \\ \cdot (\sigma^2)^{-(\frac{\nu_0}{2}+1)} \exp\left(-\frac{1}{2}\cdot\nu_0 s_0^2\cdot\sigma^{-2}\right) \\ \cdot \rho^{(Gr-1)} \exp\left(-\frac{\rho}{s}\right). \end{aligned} \tag{3.13}$$

Based on this joint distribution, it is straightforward to derive the conditionals for each parameter. We first consider the blocked sampling of $\boldsymbol{\beta}$ and σ^2, by first sampling from the marginal distribution of $p(\sigma^2|\bullet)$ and then sampling from $p(\boldsymbol{\beta}|\sigma^2, \bullet)$ where $\bullet$ denotes all the other variables. The resulting distributions have the standard form:

$$p(\sigma^2|\Lambda, \rho, \alpha, X, \boldsymbol{y}) = \text{IG}\left(\sigma^2|\frac{\nu_0+n}{2}, \frac{\nu_0 s_0 + \boldsymbol{y}^t\boldsymbol{y} - \hat{\boldsymbol{\beta}}^t(X^tX+\Lambda^{-1})^{-1}\hat{\boldsymbol{\beta}}}{2}\right), \tag{3.14}$$

where IG is the inverse-gamma distribution and $\hat{\boldsymbol{\beta}} = X^t\boldsymbol{y}$.

$$\begin{aligned} p(\boldsymbol{\beta}|\sigma^2, \Lambda, \rho, \alpha, X, \boldsymbol{y}) = N(\boldsymbol{\beta}|\tilde{\boldsymbol{\mu}}, \sigma^2\widetilde{\Sigma}) \\ \text{with } \widetilde{\Sigma} = (X^tX+\Lambda^{-1})^{-1}, \text{ and } \tilde{\boldsymbol{\mu}} = \widetilde{\Sigma}X^tX\hat{\boldsymbol{\beta}}, \end{aligned} \tag{3.15}$$

where Λ is a diagonal matrix consisting of λ_g^2's as diagonal elements, with each λ_g^2 repeated p_g times as shown in Figure 3.2:

$$\Lambda = diag(\underbrace{\lambda_1^2, \ldots, \lambda_1^2}_{p_1 \text{ replications}}, \ldots, \underbrace{\lambda_G^2, \ldots, \lambda_G^2}_{p_G \text{ replications}}). \tag{3.16}$$

The posterior conditional distribution of λ_g^2 is the generalized inverse Gaussian:

$$\begin{aligned} p(\lambda_g^2|\boldsymbol{\beta}, \sigma^2, \rho) &\propto (\lambda_g^2)^{-\frac{p_g}{2}+\alpha\frac{p_g+1}{2}-1} \exp\left[-\frac{b_g}{2\lambda_g^2} - \lambda_g^2\frac{a_g}{2}\right] \\ &= \text{GIG}\left(\alpha\frac{(p_g+1)}{2} - \frac{p_g}{2}, a_g, b_g\right), \end{aligned} \tag{3.17}$$

where $a_g = p_g\rho$ and $b_g = \|\boldsymbol{\beta}_g\|_2^2/\sigma^2 \;\forall\; g \in \{1, 2, ..., G\}$.

For the sampling of (ρ, α), sampling of ρ and α based on their individual posteriors conditioned on each other is avoided, since this results in a slow mixing of the Markov

chain due to a high correlation between samples from the two conditionals. To overcome this issue, we propose doing block sampling over the variables by first sampling α based on the marginal posterior distribution and then sampling ρ conditioned on α. The posterior conditional distribution of ρ given α results in a gamma distribution:

$$p(\rho|\alpha, \bullet) \propto \text{Gamma}\left(\rho|\alpha(s + \sum_{g=1}^{G} p_g^{'}) + 1, \sum_{g=1}^{G} \frac{p_g \lambda_g^2}{2} + q\right), \tag{3.18}$$

and the marginal of α is derived based on the work in [28]:

$$p(\alpha|\bullet) \propto \frac{t^{\alpha-1}}{\Gamma(\alpha)^r} . \prod_{g=1}^{G} \frac{(\lambda_g^2)^{p_g^{'}\alpha}}{\Gamma(p_g^{'}\alpha)} . \frac{\Gamma(\alpha(s + \sum_{g=1}^{G} p_g^{'}) + 1)}{(\sum_{g=1}^{G} \frac{p_g \lambda_g^2}{2} + q)^{\alpha(s+\sum_{g=1}^{G} p_g^{'})+1}}. \tag{3.19}$$

This marginal distribution is a non-standard distribution and is complicated to sample from. Hence sampling is done by discretizing the distribution over a range of values.

We now construct an MCMC sampling algorithm which samples values of the parameters based on all the above posterior conditional distributions. Details are given in Algorithm 1.

Algorithm 1 Gibbs Sampling for Grouped Variable Selection

1: **Input:** n observations $D = (\boldsymbol{x}_i, y_i)$.
2: **Initialize:** Parameters $\boldsymbol{\beta}, \sigma^2, \rho, \Lambda, \alpha$.
3: Draw samples from the posterior of joint distribution $p(\boldsymbol{\beta}, \sigma^2, \rho, \Lambda, \boldsymbol{\alpha}|D)$ by drawing from the conditionals.
4: **for** $m = 1$ to BayesIter **do**
5: Sample α $|\boldsymbol{\beta}, \sigma^2, \Lambda, D$ - from a discretized version of the distribution given in eqn. 3.19.
6: Sample ρ $|\boldsymbol{\beta}, \sigma^2, \Lambda, \alpha, D$ - from a gamma distribution given in eqn. 3.18.
7: Sample Λ $|\boldsymbol{\beta}, \sigma^2, \rho, \alpha, D$ - from a generalized inverse Gaussian distribution given in eqn. 3.17.
8: Sample $\boldsymbol{\beta}, \sigma^2$ $|\rho, \Lambda, \alpha, D$ - σ^2 is sampled from an inverse-gamma distribution (eqn. 3.14) and $\boldsymbol{\beta}$ conditioned on σ^2 from a multivariate normal distribution (eqn. 3.15).
9: **end for**

The Gibbs sampler for the Lasso has been shown to possess geometric ergodicity in [8] which might indicate a rapid convergence of the sampler. After running the Gibbs sampler, the generated samples can be used to estimate various quantities like the expectation and variances of the regression coefficients. We also quantify the significance of a coefficient by looking at its probability values associated with credibility intervals $[0, \infty)$ or $(-\infty, 0]$ based on whether the coefficient is positively or negatively significant. Hence, a value closer to 50% indicates non-significance whereas a value closer to 100% strongly indicates significance. Based on thresholding of these values, we obtain a sparse subset of significant variables.

3.6 Experiments

In this section, we illustrate the working of the MCMC algorithm for Bayesian grouped variable selection with two experiments. The first involves using the Bayesian Group-Lasso for variable selection in categorical variables. The second one uses the same algorithm for the special case of Bayesian Lasso in order to address the issue of correlated variables as discussed in the previous chapter.

3.6.1. Categorical Variable Selection

For this experiment, the data was generated based on the linear regression model with Gaussian likelihood. The size of the dataset is $n = 100$. For input variables, we use 30 categorical variables which can take values $\{0, 1, 2\}$. Using polynomial contrast coding, each categorical variable is transformed into two dummy variables. Hence the problem of selecting significant categorical variables is transformed into selecting groups of dummy variables which represent the respective original categorical variables. After transforming into dummy variables, total number of variables $d = 60$. The coefficients for the dummy variables which represent the categorical variables $(x_4, x_7, x_{15}, x_{25})$ are set as significant (β values 10.0) and the coefficients for rest of the variables with low significance is set to 2.0. We set the hyper-parameter values for $n_0 = n$ and $s_0^2 = 0.5$. Although we defined a joint prior over (ρ, α), it results in a peaked posterior for α and hence for experiments, it is reasonable to set the value of α based on the desired sparsity of the solution and define a conjugate gamma prior distribution for ρ. For this experiment, we set $\alpha = 1$, which represents the special case of Bayesian Group-Lasso. For setting the hyper-parameters for the distribution over ρ, we use the criterion that roughly 1% of the λ_g^2 values should exceed $5 \cdot \text{median}(\boldsymbol{\lambda}^2)$. In this experiment, we set the shape and scale values as 200 and 0.1 respectively.

The Gibbs sampler was run for 5000 iterations with a burn-in period of approximately 100 iterations. A sample trace plot of a significant regression coefficient is shown in Figure 3.6. Figure 3.7 shows the results of the toy experiment as a box plot. The true groups are clearly visible in the box plot. To perform variable selection, we plot the significance values based on the credible intervals as mentioned in the previous section. Using 95% probability as the threshold for selection, we recover the truly significant groups. Figure 3.8 shows the significance plot along with the threshold for variable selection.

3.6.2. Correlated Variables

As mentioned in the previous chapter, an issue with the Lasso is that whenever there exists a group of significant and highly correlated predictor variables, the Lasso tends to select only one variable from the group. Using a simulation, we show how the Bayesian Lasso (which is a special case of the Bayesian Group-Lasso, where $p_g = 1 \ \forall \ g$), is still able to recover all the significant variables even with highly correlated predictor variables. For this purpose, we simulated data $d = 50$ and $n = 100$. The significant variables were chosen to be $(x_4, x_{14}, x_{30}, x_{37}, x_{45})$ and the corresponding β_i values were set to 1.0. The rest

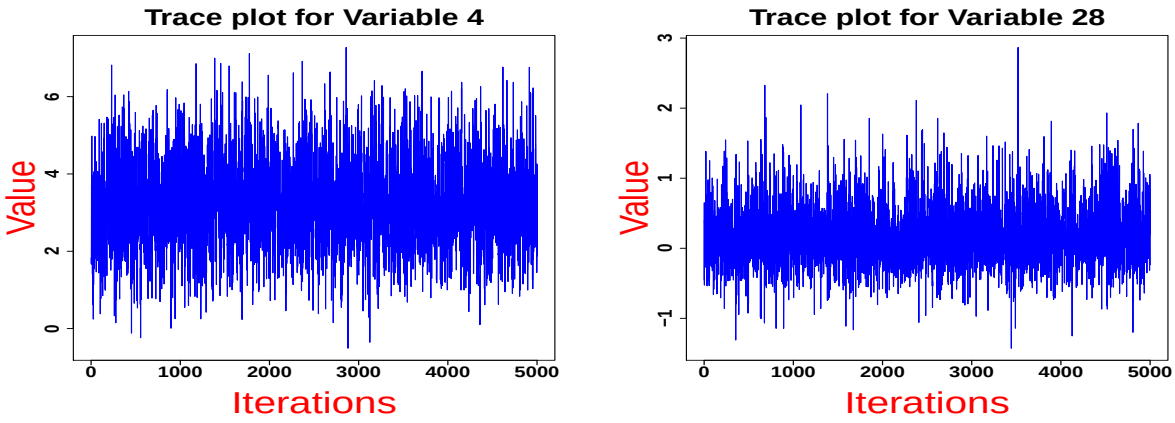


Figure 3.6.: The figure shows the trace plot for the one of the dummy variable coefficients corresponding to the original categorical variables x_4 (left) and x_{28}(right). We see that the Gibbs sampler converges very quickly and the burn-in period is hardly $50 - 100$ iterations.

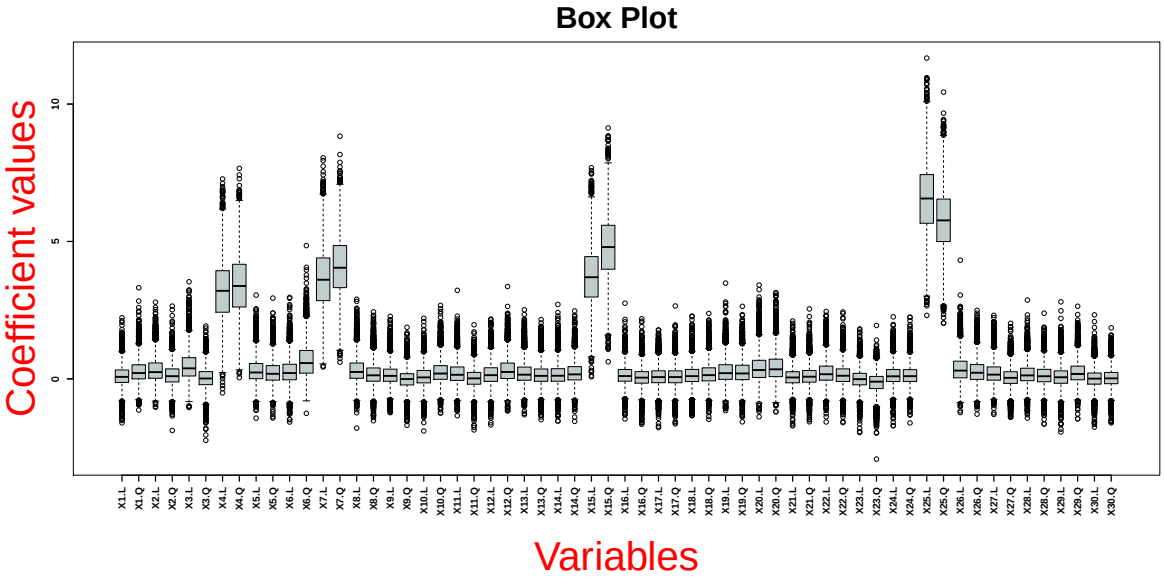


Figure 3.7.: Results of the toy experiment with a least squares model is shown in the form of a box plot of the regression coefficients. We see that the groups corresponding to the categorical variables x_4, x_7, x_{15} and x_{25} are clearly identified as significant.

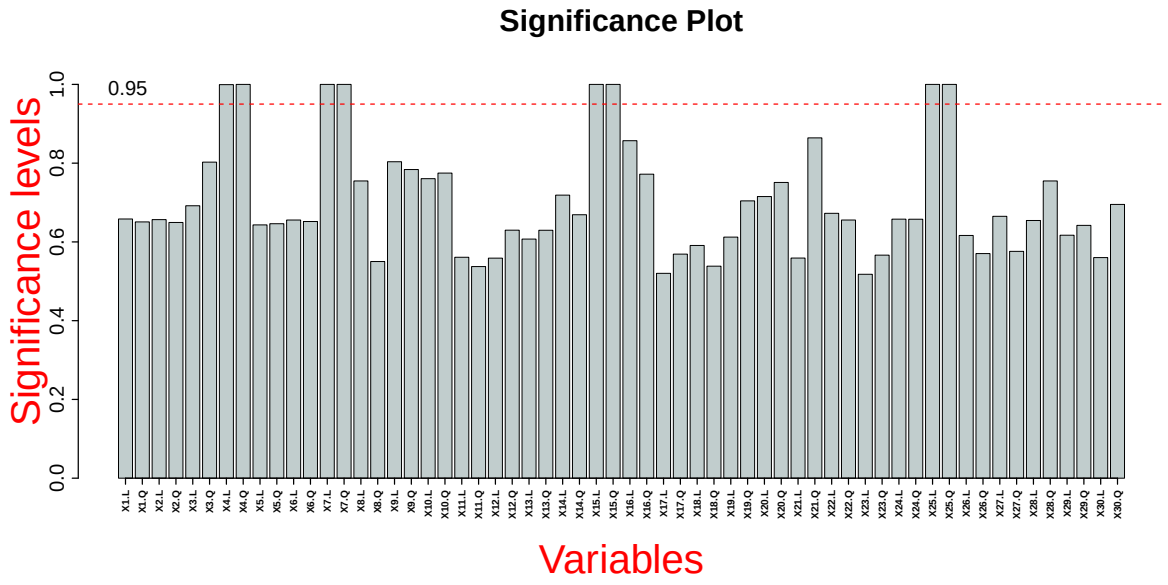


Figure 3.8.: A selection of the variables is done by plotting the significance of the coefficients based on credibility intervals. For this experiment a threshold of 95% is used and the maximum significance value of each group is used for the final selection of a variable.

of the regression coefficients were set to 0. The data was generated using a multivariate normal distribution, with variables x_4 and x_{30} being highly correlated. Using this data, we applied both the standard Lasso and the Bayesian Lasso to identify the significant variables. In the standard Lasso, as expected, one of the correlated variables (x_{30}) is not clearly identified as significant based on the solution path obtained. On the other hand, the Bayesian Lasso successfully identified all the truly significant variables including the correlated ones. The results are shown in Figure 3.9.

3.7 Summary

In this chapter, we defined the grouped-variable selection problem in the context of linear regression in the form of the Group-Lasso. We looked at the existing optimization view of the problem and then using the Bayesian Lasso as motivation, we constructed a sparsity inducing prior for grouped regression coefficients. We showed how this relates to the Group-Lasso in the negative logarithm space. Further, we observed that in the Lasso case, producing sparser solutions also results in global shrinkage of coefficients which may affect predictive performance. We extended our model further to include an extra parameter which provides additional flexibility in defining the level of sparsity desired in the solution without compromising too much on the scale of the regression coefficients in the posterior. The inference was carried using Gibbs sampling. Through simulated experiments, we looked at how samples are generated and then used to estimate various quantities like the

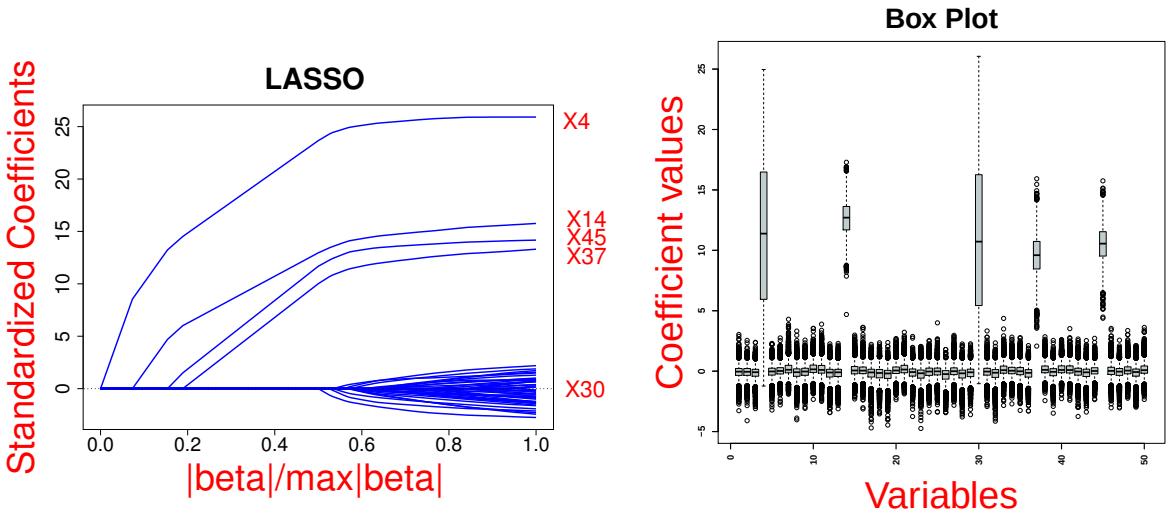


Figure 3.9.: **Left:** The solution path from the standard Lasso using the LARS package. We observe that the variable x_{30} is not clearly identified as significant. **Right:** Using the same data, the Bayesian Lasso was used to produce the box plot as shown here. We observe that both the correlated variables can be clearly identified as significant in this case.

mean, variance and significance levels. Thresholding was used to produce a point estimate, i.e. the set of selected groups of variables.

So far we have built a powerful Bayesian framework for grouped variable selection in the context of linear regression with a Gaussian likelihood. In the next chapter, we go beyond a Gaussian likelihood model and look at some other generalized linear models which can also be incorporated into this current framework with minimal changes.

4

Network Inference with Generalized Linear Models

4.1 Beyond Regression

So far, we have built an omnibus Bayesian framework for grouped-variable selection which can produce different estimates related to the posterior distribution of the regression coefficients. However, the entire analysis was done in the context of linear regression where the response variables are real numbers. But there exist other widely encountered application scenarios where the response variables are of different types. Some common examples are classification problems involving binary response variables and count data involving whole number response variables. In order to cater to a wider variety of applications scenarios, we need to look at possible extensions to the framework which would allow the inclusion of other data types for the response variables. In this chapter, we address this need by using the generic component structure of generalized linear models. We show how this generic component structure can be easily incorporated into our existing framework for linear regression for specific models with a minimal change to the MCMC algorithm.

Apart from this extension, we also extend the features from individual variables to higher-order interactions which leads us to an alternate view of variable selection. Such higher-order interactions can be represented in the form of a hypergraph where the nodes denote the variables and an edge between two variables denotes an interaction between them. This leads to the interpretation of the variable selection problem as a sparse hypergraph inference problem. We illustrate this in detail through real-world experiments in the context our extension of the Bayesian framework to generalized linear models.

These extensions are especially motivated by certain biological applications that were encountered during the course of this work. As we shall see in the experiments section, this extended model was applied to microarray datasets from breast cancer tissue samples in order to find novel "compound" bio-markers. The word compound denotes a group of associated bio-markers.

4.2 Generalized Linear Models

As described in section 2.2, the applicability of linear regression models can be broadened by extending them to generalized linear models. The GLM was described in terms of three components, the random component, systematic component and the link function. It was further generalized to the random intercept model which is illustrated in Figure 4.1.

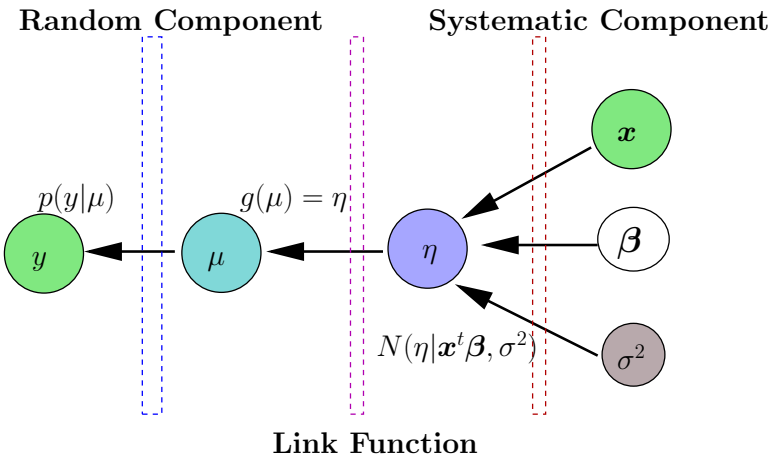


Figure 4.1.: Dependency structure of a random intercept model (i.e. a GLM with a stochastic systematic component). The dotted blocks represent the different components of the model.

Extension to Bayesian Grouped-Variable Selection. Using the above described graphical model for generalized linear models, it is straightforward to describe the posterior over the regression coefficients $\boldsymbol{\beta}$ by defining a suitable prior:

$$p(\boldsymbol{\beta}, \boldsymbol{\eta}|\boldsymbol{y}, X, \bullet) = p(\boldsymbol{y}|g^{-1}(\boldsymbol{\eta}))\ N(\boldsymbol{\eta}|X\boldsymbol{\beta}, \sigma^2 I)p(\boldsymbol{\beta}|\bullet), \tag{4.1}$$

where $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)$. To combine this with grouped-variable selection, we can re-use the hierarchical model that we have defined for the prior over regression coefficients. Applying the previously defined Bayesian grouped variable selection framework to specify the prior over the regression coefficients $\boldsymbol{\beta}$, we obtain the resulting graphical model as shown in Figure 4.2. As we have seen in the previous chapter, Gibbs sampling was the choice of inference algorithm to generate samples from the posterior distribution over $\boldsymbol{\beta}$. An advantage of using Gibbs sampling is that it is possible to incorporate additional variables to the model with minimal changes. In the context of our extension to GLMs, re-using the same inference algorithm involves only a minor addition. Since the distributions over all existing variables remain the same, the only new variable to consider is $\boldsymbol{\eta}$. Hence for posterior sampling, we need to additionally consider sampling $\boldsymbol{\eta}$ given all the other variables. The sampling for all the other variables $\boldsymbol{\beta}, \Lambda, \sigma^2$ and ρ remains the same.

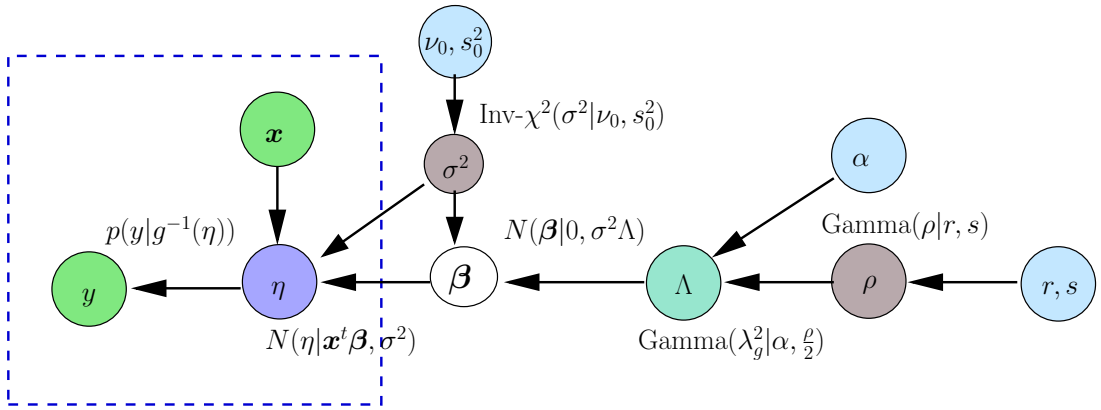


Figure 4.2.: Dependency structure of the hierarchical Bayesian Group-Lasso for a random intercept model. The changed elements are marked with a dotted rectangle indicating the change from the previously defined model for regression which was the GLM with identity link function. The rest of the model remains the same.

The sampling of $\boldsymbol{\eta}$ depends on $p(\boldsymbol{y}|\cdot)$ and the link function $g(\cdot)$ and is generically written as:

$$p(\boldsymbol{\eta}|\bullet) \propto p(\boldsymbol{y}|g(\cdot))N(\boldsymbol{\eta}|X\boldsymbol{\beta}, \sigma^2 I). \tag{4.2}$$

Hence the exact manner in which $\boldsymbol{\eta}$ will be sampled will depend on the particular GLM being analyzed along with the choice of link function. The modified Gibbs sampling algorithm for GLMs is given in Algorithm 2.

Algorithm 2 Gibbs Sampling for Grouped Variable Selection in GLMs

1: **Input:** n observations $D = (\boldsymbol{x}_i, y_i)$.
2: **Initialize:** Parameters $\boldsymbol{\beta}, \boldsymbol{\eta}, \sigma^2, \rho, \Lambda, \alpha$.
3: Draw samples from the posterior of joint distribution $p(\boldsymbol{\beta}, \boldsymbol{\eta}, \sigma^2, \rho, \Lambda, |\alpha, D)$ by drawing from the conditionals.
4: **for** $m = 1$ to BayesIter **do**
5: Sample ρ $|\boldsymbol{\eta}, \boldsymbol{\beta}, \sigma^2, \Lambda, \alpha, D$ - from a gamma distribution given in eqn. 3.18.
6: Sample Λ $|\boldsymbol{\eta}, \boldsymbol{\beta}, \sigma^2, \rho, \alpha, D$ - from a generalized inverse Gaussian distribution given in eqn. 3.17.
7: Sample $\boldsymbol{\eta}|\boldsymbol{\beta}, \Lambda\sigma^2, \rho, \alpha, D$ - from a conditional distribution based on eqn. (4.2).
8: Sample $\boldsymbol{\beta}, \sigma^2$ $|\boldsymbol{\eta}, \rho, \Lambda, \alpha, D$ - σ^2 is sampled from an inverse-gamma distribution (eqn. 4.10) and $\boldsymbol{\beta}$ conditioned on σ^2 from a multivariate normal distribution (eqn. 4.11).
9: **end for**

In this work, we discuss two widely encountered models, namely the Poisson model for count data and the binomial model for classification. We will derive the posterior

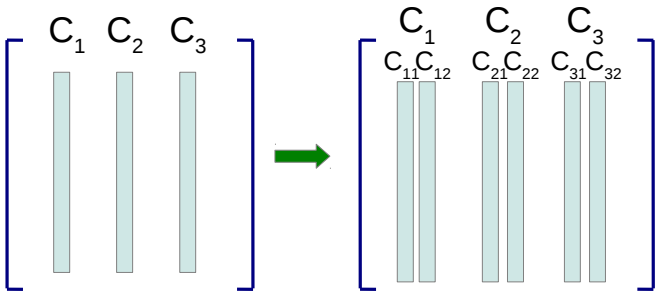


Figure 4.3.: The figure shows how a design matrix with 3 columns of categorical variables is transformed using dummy variables. Here the columns are represented by the light-blue rectangles. The resulting matrix has multiple columns per categorical variable. For example, C_{11} and C_{12} together represent the categorical variable C_1.

conditional distribution for $\boldsymbol{\eta}$ for each case separately.

Dummy Variables for Categorical Data Analysis. In this section, we will describe the usage of dummy variables for categorical data analysis and see how it motivates the grouped-variable selection problem from an individual variable selection problem. This fact is then used in subsequent biological data analysis where we frequently encounter categorical data.

Consider a data analysis problem in the context of linear regression which involves variable selection for predictors which are categorical in nature. The standard approach for such an analysis problem is to encode each categorical variable as a group of dummy variables. Various encoding techniques exist in literature like effect coding, polynomial contrast coding etc. In terms of the design matrix X, the dummy coding procedure results in introducing a group of columns for each column vector (representing one variable) as shown in Figure 4.3.

In this thesis, we use orthogonal polynomial contrast codes for dummy coding. Contrast coding is a coding procedure which models the pattern of the differences of the response variable mean for different levels or categories of a categorical variable (see [29]). Polynomial contrasts model these patterns in terms of polynomials (line, parabola etc.) and are applicable in cases when the categorical variable is quantitative in nature and the levels are equally spaced. Further, using orthogonal polynomial contrast codes have the additional benefit of making the design matrix orthogonal which provides computational benefits as we shall see later. In the next section, we will discuss the extension of the regression model to higher-order interactions and its interpretation in terms of a hypergraph.

4.3 Sparse Hypergraph Inference Problem

In this section, we look at another viewpoint with regard to variable selection which is a graphical view of the variables and the interactions between them. Consider the simple

linear regression problem as before where:

$$y = \boldsymbol{x}^t\boldsymbol{\beta} + \epsilon \;\text{ where } \epsilon \sim N(0, \sigma^2). \tag{4.3}$$

The feature space can be extended to include higher-order interaction terms (first-order and above) between all the variables where the zeroth order interaction terms are the individual variables, and the n-th order interactions represent the interactions between all combination of $(n-1)$ variables. For example, adding the first and second-order interaction terms to our regression model will result in:

$$y = \sum_i x_i\beta_i + \sum_{i,j} x_i x_j \beta_{ij} + \sum_{i,j,k} x_i x_j x_k \beta_{ijk} + \epsilon. \tag{4.4}$$

We will denote the augmented regression coefficient vector as $\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ consists of all the regression coefficient terms including the higher order interaction terms β_{ij} and β_{ijk}. The problem of sparse feature selection can now be extended to include these higher-order interactions as well. The sparseness of the augmented coefficient vector $\boldsymbol{\beta}$ (with all higher order terms) can now be represented as a hypergraph.

A hypergraph is a generalization of a graph where an edge is defined to be between any number of vertices, instead of just two as in a usual graph. We can represent the regression coefficient vector $\boldsymbol{\beta}$ in terms of a hypergraph, where the nodes represent the individual variables and edges represent the interactions. In this thesis, we consider interaction terms only upto second-order. Hence, the corresponding hypergraph will have hyperedges at the most between three nodes. Specifically, all the coefficients β_{ij} represent an edge between variables i and j and all coefficients β_{ijk} represent an edge between variables i, j and k. The size of the circles and the width of the edges denote the level of significance of the particular interaction term. A sample sparse hypergraph representing a sparse regression coefficient vector is shown in Figure 4.4.

For the rest of this document, we will use this representation for illustrating a sparse coefficient vector. The weight on the edges/nodes will denote the level of significance of the particular interaction term which will be measured in the usual manner as done in previous experiments.

4.4 Poisson Models for Contingency Tables

In section 4.2, we discussed the extension of the Bayesian grouped-variable selection to generalized linear models. We now focus on a specific GLM, namely the Poisson model for analyzing count data in contingency tables.

Contingency Tables. A contingency table represents a G-dimensional table where each dimension represents a categorical variable. Consider a contingency table τ comprising of G categorical variables $\{C_1, \ldots, C_g, \ldots, C_G\}$ with each variable C_g consisting of K_g categories. Each cell is denoted by $(v_1, v_2, \ldots, v_G)$ where v_g denotes the categorical value

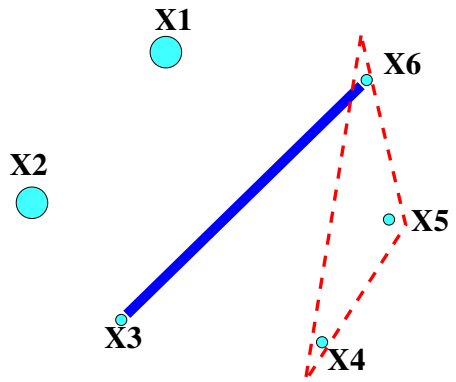


Figure 4.4.: An example of a sparse hypergraph between five variables representing the coefficient vector which has non-zero entries for $\beta_1, \beta_2, \beta_{36}, \beta_{456}$. The zeroth order interactions (i.e individual variables) are represented by circles, first-order or pairwise interactions are represented by blue lines and second-order or triplet interactions are represented by red triangular hyperedges. The size of the circles and width of the edges denote the level of significance of the particular interaction term.

corresponding to C_g for that cell. Hence a value in a cell represents the number of observations with that combination of values for the categorical variables. The total number of cells in the contingency table will be $n = \prod_{g=1}^{G} K_g$.

Based on the assumption of how the counts in cells were generated, the two different models that can be used for analyzing contingency tables are the multinomial model and the Poisson model. Assuming that the total number of counts is fixed, the observed vector of individual counts $\boldsymbol{y} = (y_1, \ldots, y_n)$ can be considered as a realization drawn from a multinomial distribution. This is the approach taken in [30] in a penalized likelihood framework. If on the other hand, we assume a sampling model in which the total number of counts itself is random and the time period for observing the counts is fixed, we arrive at the Poisson model. This sampling model is plausible for many practical situations. For example, a Poisson model can be applied to a clinical study where the counts correspond to the number of patients with certain properties visiting the hospital in a fixed time period.

The motivation for analyzing count data in contingency tables arises due to several reasons. Firstly, count data for certain compositions of categorical variables occurs frequently in practical applications, particularly in a bio-medical context like in tissue microarray data for measuring protein expression levels. Secondly, due to the introduction of dummy variables, feature selection for categorical variables is directly related to inferring sparsity on the level of groups of predictor variables. Based on these motivations, we show that choosing a specific encoding scheme for categorical variables allows us to derive a highly efficient sampling algorithm that makes full Bayesian inference practical for large-scale applications.

Construction of the Design Matrix X. Let $\boldsymbol{y} = (y_1, \ldots, y_n)$ be the observed counts in each cell of the contingency table as described above. Starting from the counts existing in the contingency tables, we describe the construction of the design matrix X, including the steps of creating the dummy variables and adding higher-order interactions.

1. *Construct a 2-dim table.* We first create a 2-dimensional table where each row corresponds to one cell of the contingency table. Hence the number of rows is n. The columns represent the G variables and hence encode the value that each variable takes for that cell. The corresponding count column vector $\boldsymbol{y}$ gives the count observed for the particular cell indexed by rows.

2. *Higher order interactions.* Along with this, we also augment our model, and hence the matrix, by adding higher-order interactions terms. Interaction terms are basically column-wise product expansions of these individual (main-effect) matrices. The augmented design matrix X can now be viewed as being composed of individual sub-matrices:

$$X = [\mathbf{1}, \underbrace{X^{C_1}, \ldots, X^{C_G}}_{\text{main effects}}, \underbrace{X^{C_1:C_2}, \ldots, X^{C_{G-1}:C_{Gd}}}_{\text{1st order interactions}}, \ldots, \underbrace{X^{C_1:\cdots:C_{Q+1}}, \ldots, X^{C_{G-Q}:\cdots:C_G}}_{\text{highest order interactions}}], \tag{4.5}$$

   where each sub-matrix encodes q-th order interaction terms and the interactions are denoted by the colon operator (:) and Q denotes the highest order interaction that is modeled. The individual variables are denoted as zeroth order, pair-wise interactions as first order and so on.

3. *Dummy coding.* As described in the previous section, categorical variables need to be encoded using dummy variables. In terms of the design matrix X, we need to replace each column g with a group of columns, using dummy variables which we denote by $\{C_g^{dm}\}$. To avoid over-parametrization, identifiability constraints are imposed on the individual sub-matrices in the form of *contrast codes* which encode a factor with K levels into a sub-matrix with $K - 1$ columns. In many practical applications, we are given *ordered* factors, i.e. ordinal variables for which a natural ordering is involved. Examples of this kind are, for instance, intensity levels that are measured in protein expression data. For such ordinal categorical variables, the use of polynomial contrast codes (see [29]) is a natural choice. These encodings employ orthogonal polynomials and have the practical advantage that the (typically huge) resulting design matrix is orthogonal, i.e. $X^t X = I$.

As a result of the above steps, we obtain groups of "dummy" variables where each group encodes an interaction term, starting from the zeroth (or single variables) to a maximum Qth order interaction terms. For the experiments that we consider in this thesis, $Q = 2$.

GLM Components for a Poisson Model. Assuming that the counts are random and were generated in a fixed time period, the standard approach for modeling count data

over a fixed period of time is Poisson regression which involves a log-linear model with a *random component* of independent terms:

$$y_i|\mu_i \sim \text{Poisson}(\mu_i) = \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}, \quad \text{for } i = 1, \ldots, n \tag{4.6}$$

where "Poisson" denotes the Poisson distribution. The link function defined on the mean μ_i is $\mu_i = e^{\eta_i}$, and the stochastic systematic component is defined as before:

$$\eta_i = \boldsymbol{x}_i^t \boldsymbol{\beta} + \epsilon_i, \ \epsilon_i \sim N(0, \sigma^2), \tag{4.7}$$

where the stochastic definition makes it possible to allow for deviations from the log-linear model. This is applicable, for example, to overdispersed Poisson models which are common in practice. Hence the updated likelihood term is as follows:

$$p(\boldsymbol{y}, \boldsymbol{\eta}|X, \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} \text{Poisson}(y_i|\exp(\eta_i)) N(\eta_i|\boldsymbol{x}_i^t \boldsymbol{\beta}, \sigma^2). \tag{4.8}$$

The main technical advantage of the Poisson model lies in the factorization over the cells, given the means μ_i (see eq. (4.6)). We shall see later that this simplifies sampling of $\boldsymbol{\eta}$ in the Gibbs sampling algorithm.

Inference Using Gibbs Sampling. In a generalized linear model, we can re-use the Gibbs sampling inference technique that we introduced previously for the Bayesian grouped-variable selection framework. For the Poisson Model, the joint distribution over all the variables can be written as:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\eta}, X, \boldsymbol{\beta}, \Lambda, \sigma^2, \rho) \propto & \prod_{i=1}^{n} \text{Poisson}(y_i|\exp(\eta_i)) N(\eta_i|\boldsymbol{x}_i^t \boldsymbol{\beta}, \sigma^2) \\ & \cdot (\sigma^2)^{-\frac{d}{2}} \prod_{g=1}^{G} \left[(\lambda_g^2)^{-\frac{p_g}{2}} \exp\left(-\frac{\|\boldsymbol{\beta}_g\|^2}{2\lambda_g^2 \sigma^2} \right) \right. \\ & \left. \cdot (\lambda_g^2)^{\frac{(p_g+1)}{2}-1} \rho^{\frac{(p_g+1)}{2}} \exp\left(-\lambda_g^2 \frac{p_g \rho}{2} \right) \right] \\ & \cdot (\sigma^2)^{-(\frac{\nu_0}{2}+1)} \exp\left(-\frac{1}{2} \cdot \nu_0 s_0^2 \cdot \sigma^{-2} \right) \\ & \cdot \rho^{(Gr-1)} \exp\left(-\frac{\rho}{s} \right). \end{aligned} \tag{4.9}$$

The use of a stochastic systematic component not only allows for deviations from the parametric model, but also greatly simplifies Gibbs sampling. This is due to the fact that conditioning on η_i, sampling of $\boldsymbol{\beta}$ and σ remains the same. Also, a key benefit in using orthogonal contrast codes is the resulting orthogonality property of the design matrix, $X^t X = I$. This makes it possible to sample from very high-dimensional models in an

efficient and numerically stable way since all the matrix inverse operations involve only diagonal matrices. The modified posterior conditionals for $\boldsymbol{\beta}$ and σ^2 are:

$$p(\sigma^2|\Lambda_d, \rho, \alpha, X, \boldsymbol{\eta}) = \text{IG}\left(\sigma^2|\frac{\nu_0 + n}{2}, \frac{\nu_0 s_0 + \boldsymbol{\eta}^t\boldsymbol{\eta} - \hat{\boldsymbol{\beta}}^t\Lambda_d\hat{\boldsymbol{\beta}}}{2}\right), \tag{4.10}$$

where IG is the inverse-gamma distribution and $\hat{\boldsymbol{\beta}} = X^t\boldsymbol{\eta}$.

$$\begin{aligned} p(\boldsymbol{\beta}|\sigma^2, \Lambda_d, \rho, \alpha, X, \boldsymbol{\eta}) &= N(\boldsymbol{\beta}|\tilde{\boldsymbol{\mu}}, \sigma^2\Lambda_d) \\ &\text{with } \tilde{\boldsymbol{\mu}} = \Lambda_d\hat{\boldsymbol{\beta}}, \end{aligned} \tag{4.11}$$

where Λ_d is a diagonal matrix consisting of values $\lambda_{dg} = \frac{1}{1+\frac{1}{\lambda_g^2}}$ with each λ_{dg} repeated p_g times:

$$\Lambda_d = diag(\underbrace{\lambda_{d1}, \ldots, \lambda_{d1}}_{p_1 \text{ replications}}, \ldots, \underbrace{\lambda_{dG}, \ldots, \lambda_{dG}}_{p_G \text{ replications}}). \tag{4.12}$$

The sampling of $\boldsymbol{\beta}$ is efficient since all the components can be sampled independently. The only addition to the Gibbs sampling algorithm is the sampling of η_i from its posterior conditional distribution:

$$p(\eta_i|\bullet) \propto \exp\left[\eta_i y_i - \exp(\eta_i) - \frac{1}{2\sigma^2}(\boldsymbol{x}_i^t\boldsymbol{\beta} - \eta_i)^2\right] \qquad \forall\ i = 1 \ldots n. \tag{4.13}$$

Since the above conditional is not of recognized form, we need to look at alternate ways of sampling from this distribution. The above conditional posterior is log-concave which makes it possible to use "black-box" sampling methods like adaptive rejection sampling. Another possibility is to use a Laplace approximation which has been used in a similar context of Poisson regression in [31]. We propose using Laplace approximation, since in practice it gives results which are almost indistinguishable from adaptive rejection sampling, along with speeding up the sampling step considerably.

The Laplace approximation (see [32])involves approximating a target distribution P with the normal distribution Q. This is done by a Taylor expansion of the un-normalized log-probability distribution around its peak and then using the first three terms for the approximation. The second term is zero since the first derivative vanishes at the peak, hence only the first and third term are included. As a result, the distribution is approximated to:

$$Q(x) = \sqrt{\frac{c}{2\pi}}\exp\left(-\frac{c}{2}(x - x_0)\right), \tag{4.14}$$

where x_0 is the mode of the distribution P and $c = \frac{\partial^2}{\partial x^2}\ln P(x)|_{x=x_0}$. Similarly, we approximate $p(\eta_i)$ from eq. (4.13) to a normal distribution $N(\eta_i|\eta_{i0}, c^{-1})$. We derive the values of η_{i0} and c by solving:

$$y_i - \exp(\eta_i) + \frac{1}{\sigma^2}(\boldsymbol{x}_i^t\boldsymbol{\beta} - \eta_i) = 0, \tag{4.15}$$

where η_{i0} is the solution to this equation. With this value, c is calculated as:

$$c = \exp(\eta_{i0}) + \frac{1}{\sigma^2}. \tag{4.16}$$

Hyperparameter Selection. A side effect of using the Laplace approximation for η_i is a good intuition about reasonable priors on σ^2. Such a prior should be roughly centered around the reciprocal value of the average of all counts, see [31] for details. In our implementation we use this rule of thumb by setting s_0^2 in the Inv-$\chi^2(n_0, s_0^2)$ prior on σ^2 to $1/(1 + \text{median}(\boldsymbol{y}))$.

4.4.1. Application to Breast Cancer Studies

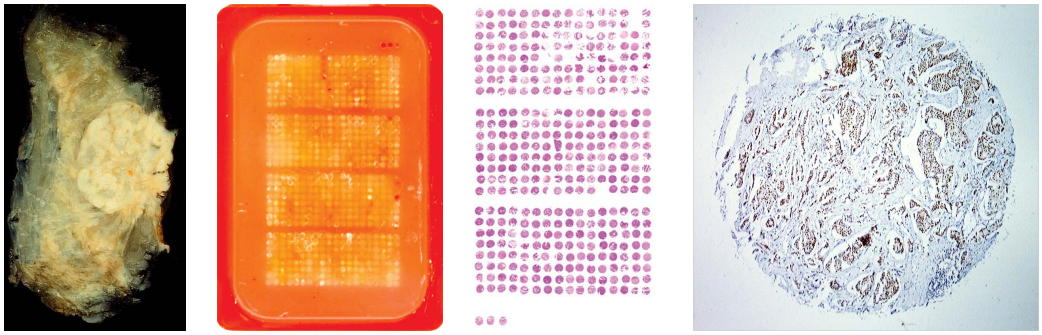
Breast Cancer and Immunohistochemistry. In western societies, breast cancer is one of the leading causes of tumor-induced death in women. Despite improvements in the identification of prognostic and predictive parameters, novel biomarkers are needed to improve patient risk stratification and to optimize patient outcome. Furthermore, the identification of molecules that are differentially regulated during tumorigenesis may lead to the development of personalized medicine for patients.

Recently, independent research groups were able to identify five distinct gene expression patterns which are (i) highly predictive for the patients' prognosis and (ii) may reflect the biological behavior better compared to established parameters. According to this model, a basal as well as two distinct luminal-like expression patterns in addition to a **her2** (**ERBB2**) over-expressing and a normal breast-like group could be distinguished [33].

Even though results from mRNA expression profiling are very convincing, there are still some limitations to its clinical application due to the high costs. Several studies have shown that biologically distinct classes of breast cancer as defined by mRNA expression analysis can also be identified with a cost efficient technique called immunohistochemistry [34, 35]. Definitions of the basal phenotype by the former and other groups using different cytokeratin antibodies (e.g. anti-**CK5/6**) prove to be robust and allow the identification of this tumor type on a routine basis.

Tissue Microarrays with Immunohistochemistry. Immunohistochemistry is a cost effective technique which is used for detecting specific antigens (e.g., proteins) in the cells of a tissue sample by exploiting the principle of antibodies binding specifically to antigens in biological tissues. An antibody-antigen interaction can be visualized in various ways like immunofluorescence and immunoperoxide staining. Immunohistochemical staining is widely used in the diagnosis of abnormal cells such as those found in cancerous tumors.

In the **tissue microarray** (TMA) technology, $0.6mm$ tissue cylinders are punched from primary tumor blocks of hundreds of different patients and subsequently embedded into a recipient tissue block. Slices of these tissues are then stained immunohistochemically to detect specific antigens (see Figure 4.5 for details). Sections from such array blocks can then be used for simultaneous *in situ* analysis of thousands of primary tumors on DNA, RNA, and protein level. The high speed of arraying, the lack of a significant damage to



(a) (b) (c) (d)

Figure 4.5.: (a) to (d) shows a flow of the process of tissue microarray analysis. First the primary samples are taken from a cancerous tissue (a breast tissue in this case as shown in (a)). Then $0.6mm$ tissue cylinders are punched from the primary tumor blocks of different patients and arrayed in a recipient paraffin block (b). Slices of $0.6\mu m$ are cut off the paraffin block and are immunohistochemically stained (c). These slices are scanned and each spot represents a different patient. Image (d) depicts the TMA spot from a single patient with breast cancer stained with an antigen.

donor blocks, and the regular arrangement of arrayed specimens substantially facilitates automated analysis. The TMA technology promises to significantly accelerate studies seeking for associations between molecular changes and clinical endpoints [36].

In the present study involving cancerous breast tissue samples, intensity levels of the following immunohistochemical markers in tissue samples have been measured utilizing the TMA technology:

1. **er** - estrogen receptor.
2. **KPNA2** - karyopherin-alpha-2.
3. **CK5/6** - anti-cytokeratin.
4. **Collagen-6** - fibrous structural protein.
5. **Claudin-7** - membrane-associated tetraspanin protein.
6. **ITIH5** - inter-α-trypsin inhibitor.
7. **her2** - the human epidermal growth factor receptor.

Experimental Design. After histopathological grading of tumors according to [37], patients were divided into a low-risk (grade 1-2) and a high-risk (grade 3) group. The overall goal of this experiment was to identify differences in interaction patterns of marker proteins between these groups. The Kaplan-Meier survival analysis in Figure 4.6 shows that

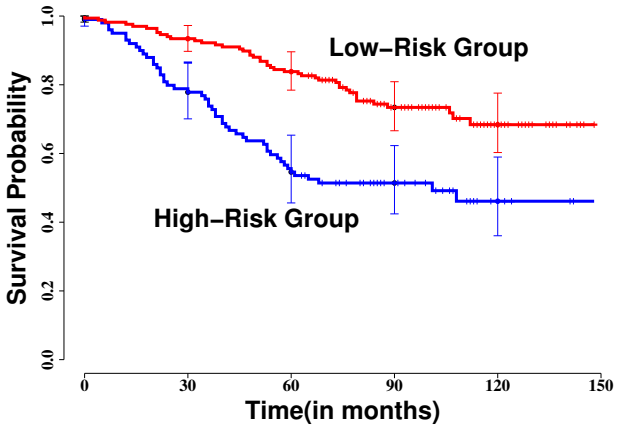


Figure 4.6.: Kaplan-Meier curves regarding overall survival for the low-risk (upper red curve) and the high-risk group (lower blue curve) of breast cancer patients. Error bars define standard 95% confidence intervals.

the split chosen is meaningful in the sense that the survival patterns of the two groups differ significantly. This observation was corroborated by the analysis of mean protein expression levels in the two groups (Figure 4.7). As expected for the low-risk class of patients (grade 1-2), there was marked estrogen receptor (**er**) expression, whereas **CK5/6** and **KPNA2** were negative.

Data Analysis and Interpretation. Having identified a meaningful subdivision into patient groups, our goal was to identify the significant interaction patterns in each group. The observed expression of each of the proteins was represented as a factor with 3 levels ("low", "intermediate", "high"). The resulting contingency tables were separately analyzed for each group with our Poisson based grouped-variable selection model. Interaction terms up to the second order (i.e. individual variables, pair-wise interactions and triplet interactions) were analyzed. In terms of the hypergraph view of the problem, the goal was to infer a sparse hypergraph, with hyperedges containing upto 3 nodes. One million Gibbs samples were drawn, the burn-in phase contained the first 200,000 samples, and every 25th of the remaining samples was used for computing the posterior densities. Due to our efficient algorithm, it took less than one hour to compute one million samples on a standard computer. Trace plots indicated that the convergence of the Markov chain was not really an issue in this experiment (see Figure 4.9 for an example), an observation which is corroborated by a length control diagnosis according to [38] indicating that the necessary burn-in-period is probably $\ll 1000$.

In the low-risk group, the following interaction terms appeared to be highly significant: the two main effects of **KPNA2** and **CK5/6** expression, the first-order interaction **KPNA2**:**CK5/6** and the second order interaction **KPNA2**:**CK5/6**:**her2**, see Figure 4.8. Interpreting high-order interaction terms can be a complex problem. A close analysis of the contrast codes and the sign of the regression coefficients showed, however, that all these interaction terms explain observed counts by either marginal or joint

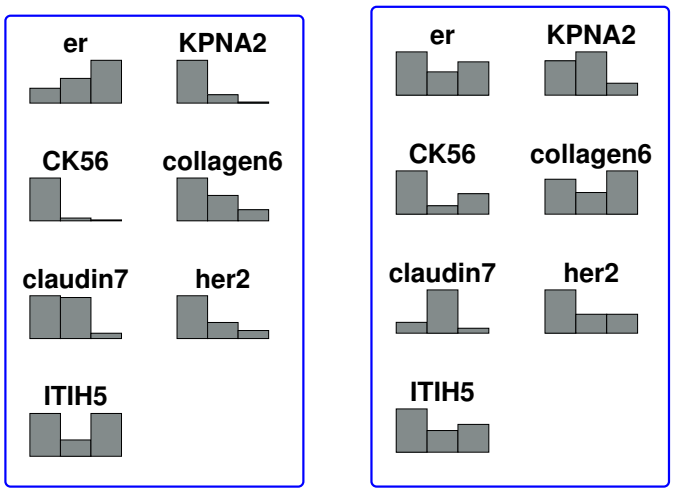


Figure 4.7.: Distribution of protein expression levels. The expressions are divided into 3 levels corresponding to the three bins shown which represent "low", "intermediate", "high" expression values. The distribution of expression levels is shown for the low-risk group (left) and the high-risk group (right).

negative immunoreactivity for **KPNA2**, **CK5/6** and **her2**, respectively. The high-risk group showed a distinctly different interaction pattern which is dominated by the main effect of **claudin7** expression, the interaction **claudin7**:**KPNA2** and the (weaker) interaction **er**:**claudin7**:**ITIH5**. Again, looking into the contrast codes and the signs, we concluded that the main effect of **claudin7** explains the counts by an over-represented "intermediate" bin and both under-represented "low" and "high" bins. The interaction term **claudin7**:**KPNA2** explains counts mainly by a joint "intermediate" expression.

These interaction patterns are in line with known gene expression patterns of breast cancer. **Claudin7** is a known intercellular adhesion molecule. As expected, the interaction pattern of high-risk tumors (grade 3) was dominated by loss of expression of the tight junction protein **claudin7**. Non-high grade breast cancers (grade 1-2) were mainly hormone-receptor positive, and negative for the high-grade markers **KPNA2**, **CK5/6** and **her2**. In a study by Dahl et al. [39], high rates of **KPNA2** expression were significantly associated with positive **TP53** and **her2** immunoreactivity and a high proliferation index. Besides **CK5/6**, **KPNA2** seemed to be characteristic of the basal-like subtype of breast cancers, possibly representing a different clinical entity of breast tumors, which is associated with shorter survival times and a high frequency of **TP53** mutations. Overexpression of the **her2** protein is also a well-known prognostic factor associated with poor survival in breast cancer, which also was found for the **her2**-positive group defined by Sorlie et al. [40].

Control Experiments. In order to compare our results with other analysis methods we conducted two control experiments. For the low-risk group, Figure 4.10 shows the "solution path" computed by the non-Bayesian analogue of our method, the standard Group-

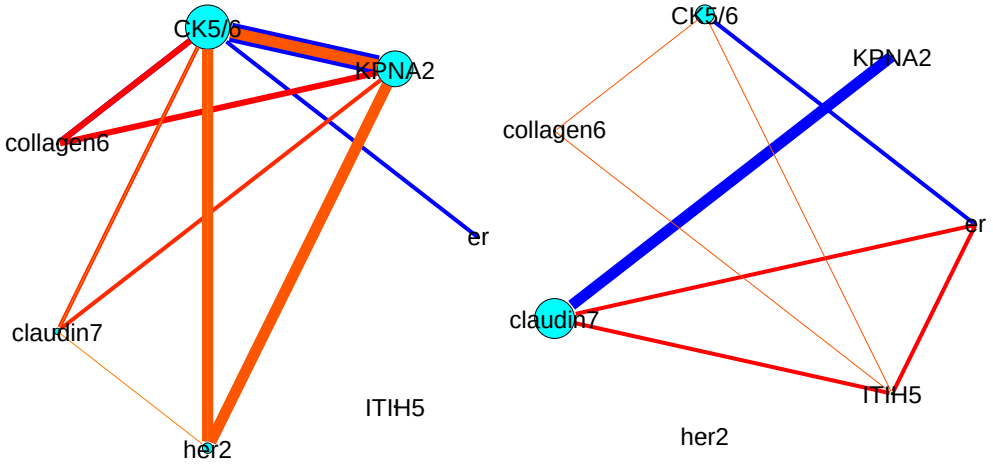


Figure 4.8.: Identified interaction patterns for the low-risk group (left) and the high-risk group (right). The size of the circles indicates the estimated significance of the main effects. For instance, the largest circle for **CK5/6** means that more than 95% of the posterior samples are negative. Correspondingly, the linewidth of the interactions (blue lines: 1st-order, reddish triangles: 2nd-order) indicates their significance, see Figure 4.9 for an example of a significance plot for an interaction.

Lasso with Poisson likelihood. We used the algorithm described in [12]. The solution path shows the evolution of the individual group norms when relaxing the constraint κ, see eq. (3.2). The plot indicates that the main effects **CK5/6** and **KPNA2** and the interactions **KPNA2**:**CK5/6** and **KPNA2**:**CK5/6**:**her2** have a dominating role, which is in perfect agreement with our results. At the same time, the more diffuse picture for large constraint values $\kappa > 100$ together with the difficulty of defining meaningful variance estimates effectively demonstrates the inherent interpretation problems of classical Group-Lasso solutions.

The test for uniqueness/completeness of solutions proposed in [12] reveals another problem: for any reasonable numerical tolerance parameter in the optimization process, the solutions found by the Group-Lasso are probably not uniquely identifiable. For constraint values $\kappa > 90$ there is an increasing amount of inactive (i.e. zero norm) groups that might become active in alternative solutions which are ϵ-close (in terms of likelihood) to the found "optimal" solution. This problem might be viewed as another strong argument for following the Bayesian paradigm of averaging over Group-Lasso solutions, instead of focusing on a single (penalized) maximum likelihood solution.

For a second control experiment we used the same data to estimate a Bayesian network. Concerning the identification of interactions, the main technical differences to our Bayesian Group-Lasso model are the restriction to a graph (instead of a hypergraph), and the use of directed edges which in some cases can be used for inferring causal relations. We used the *deal*-Package [41, 42] that finds the topology on the basis of the network score which is basically the log of the joint probability of the graph and the data. In the resulting

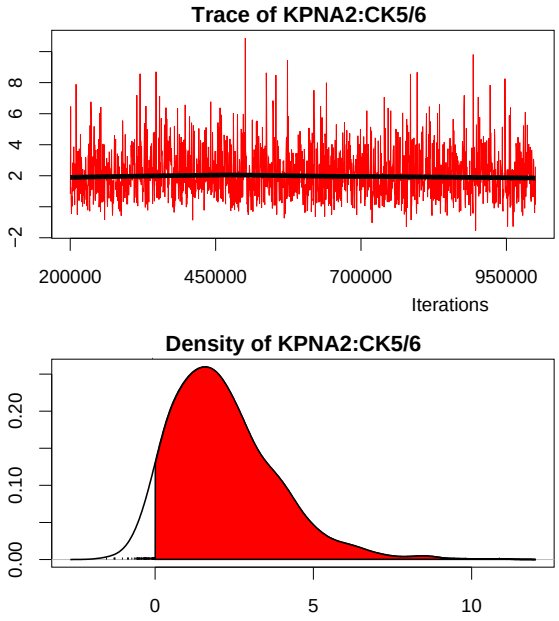


Figure 4.9.: Example trace plot for the very strong 1st-order-interaction **KPNA2**:**CK5/6** in the low-risk group, and moving average (upper panel). Corresponding posterior density (lower panel). More than 90% of the samples exceed zero (red area).

analysis, we observe many similarities between the Markov blankets from the Bayes nets and from our model. On the other hand, neither the network topology nor the direction of edges seems to be very stable. Among the top-scoring models many variants of the network have almost indistinguishable scores. Most of these fluctuations concern the dependencies between the three variables **KPNA2**, **claudin7** and **her2**, see Figure 4.11 for an example. The clear identification of a second-order interaction between these three variables in our model (the bold red triangle in the left panel of Figure 4.8) might be interpreted as a strong advantage of explicitly modeling high-order interactions in a hypergraph.

In summary, through these experiments, we have demonstrated the usefulness of the Poisson sparse regression framework for detecting novel compound-biomarkers by extending the analysis to higher order interactions.

4.5 Binomial Model for Classification

In this section, we look at yet another example of a commonly used generalized linear model, namely the binomial model for classification. Specifically we will analyze the case of a two-class classification problem via the Bernoulli model. For this purpose, as in the Poisson regression case, we will utilize the generalized linear model extension to our framework, which in this case is the Bernoulli regression model. The response variable y for a two-class classification problem is binary, where $y \in \{0, 1\}$.

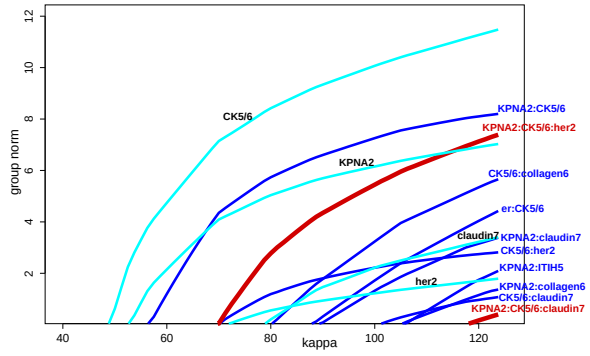


Figure 4.10.: The figure shows the comparison of results with the optimization based Group-Lasso method. The experiment was executed for the low-risk group and the result is displayed based on the evolution of group norms ("solution path") which is obtained by relaxing the κ constraint in the standard Group-Lasso with Poisson likelihood.

GLM Components for Binomial Regression. The random component of a two-class binomial regression model is defined as:

$$p(y_i|\mu_i) = \text{Bernoulli}(y_i|\mu_i), \qquad \text{for } i = 1, \ldots, n \tag{4.17}$$

where $\text{Bernoulli}(y_i|\mu_i) = \mu_i^{y_i}(1-\mu_i)^{(1-y_i)}$. The link function can be defined in multiple ways, the standard choices are the logit and probit functions (see [1]). We choose the probit link function since the computations involved for sampling from the posterior conditional distribution are straightforward. The probit link function is defined as:

$$\mu_i = \Phi(\eta_i), \tag{4.18}$$

where Φ denotes the cumulative distribution function of a standard normal distribution:

$$\Phi(\eta_i) = \frac{1}{2\pi}\int_{-\infty}^{\eta_i} \exp\left(-\frac{t^2}{2}\right) dt. \tag{4.19}$$

As before, the auxiliary variable $\boldsymbol{\eta}$ forms a part of the stochastic systematic component with $\eta_i \sim N(\boldsymbol{x}_i^t\boldsymbol{\beta}, \sigma^2)$. Similar to the Poisson case, this model can be combined with the Bayesian grouped-selection framework. All the other posterior conditional distributions are intact and the only new posterior conditional to be looked at is the one involving $\boldsymbol{\eta}$. The posterior conditional distribution for $\boldsymbol{\eta}$ factorizes into η_i for each i, where the distribution takes the form of a truncated normal distribution:

$$\eta_i \sim N(\boldsymbol{x}_i^t\boldsymbol{\beta}, 1)\begin{cases} -\infty < \eta_i < 0 & \text{if } y_i = 0 \\ 0 < \eta_i < \infty & \text{if } y_i = 1 \end{cases}, \tag{4.20}$$

which can be sampled in a very efficient manner (see [43]) with existing statistical library functions. Additionally, all the previous extensions regarding categorical variables (i.e. ad-

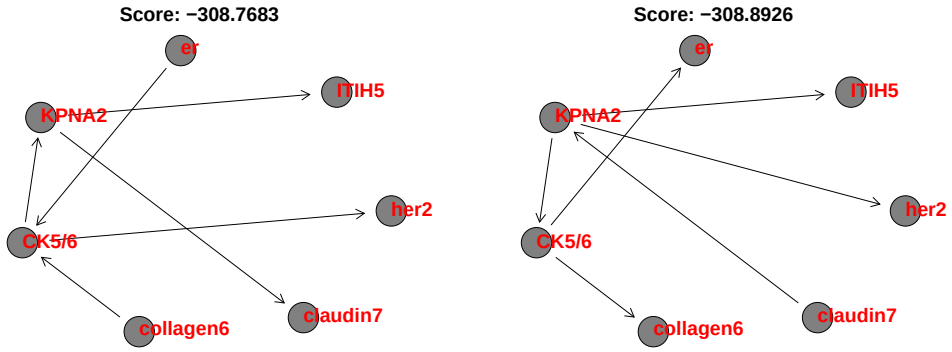


Figure 4.11.: Two examples of the top-scoring Bayes nets with almost indistinguishable scores, showing typical variations in topology. For quantifying the score differences, we made a perturbation experiment: when 10% of the observations are randomly left out, the standard deviation of the individual maximum scores is $\sigma \approx 2.2$. More than 50 different networks in the unperturbed problem lie within the highest one-σ region.

dition of dummy variables) and higher-order interactions can be applied to this model as well. We now look at the biological application of splice site detection to showcase the workings of both the binomial model and the previously discussed Poisson model.

4.6 Application to MEMset Donor Dataset

In biology, with respect to analyzing DNA sequences to find genes, it is very important to be able to recognize splice sites. Splice sites are regions in the DNA which separate coding (exons) and non-coding (introns) regions. In particular, the $5'$ end (starting point) of an intron is called the donor splice site and is analyzed in this section. For this purpose, the MEMset Donor dataset `http://genes.mit.edu/burgelab/maxent/ssdata/` is used which consists of 8415 true and 179438 false human donor splice sites data. For the analysis done in this section, the data was balanced (see [22]) in both datasets so that both have an equal size of 8415. Each instance of data consists of a sequence of DNA of length 7 within a window of the splice site which consist of the last 3 positions of the exon $(-3, -2, -1)$ and first four positions $(2, 3, 4, 5)$of the intron (see Figure 4.12). Hence these are strings of length 7 comprising of 4 characters {A, C, T, G} which represent the four nucleic acids **Adenine**, **Cytosine**, **Thymine** and **Guanine** (see [44] for details). Figure 4.13 shows the distribution of {A, C, T, G} in all the window positions in both true and false splice site datasets. Apart from the main effects, the data is extended further to include 1st order(pairwise) and 2nd order(triplet) interactions. Each interaction term is then coded with dummy variables using a polynomial contrast code giving rise to a 16384×1156 design matrix.

A Poisson model applied to contingency tables was used to analyze the interactions in both true and false splice sites separately. Figure 4.14, shows the difference in the interaction patterns of true and false splice sites. In particular, we observe a very strong

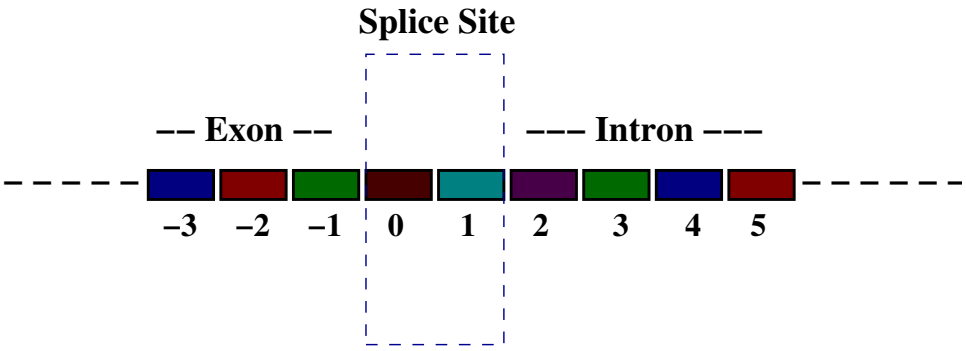


Figure 4.12.: Illustration of a splice site within a portion of a DNA. A splice site is a set of positions which separates an exon and an intron. The splice site is marked as position 0 and 1. The positions after the splice site is marked with increasing numbers $2, 3$ and so on, while the positions before the splice site are marked with decreasing numbers starting with $-1, -2$ and so on.

2nd order intra-region interaction between window positions (2:3:4) in true splices sites, which is completely missing in the case of false splice sites. Interestingly, we also observe in the true case a strong 1st order *inter-region* interaction between (-1:2), which are the last position of the exon and first position of the intron respectively which conforms to what one would hope to expect with such a sequence related pattern. This particular observation also validates the assertion made in [12], which does not find the inter-region interaction as important, but shows that inter-region interactions may have a role in solutions with the same (or ϵ-close) likelihood.

A second experiment was performed with this data in order to infer the significant interaction patterns in the context of classification of a given sequence as true or false splice site. The binomial model with probit link function was used for this purpose. Figure 4.15 shows the significant interaction patterns which help in differentiation between true and false splice sites. Apart from observing some patterns similar to the first experiment, we also observe a strong 2nd order *inter-region* interaction between (-1:3:4), which again emphasizes the importance of long range interactions in this classification task. This observation, in particular, shows the ability of the model to address the issue raised in [12], regarding the non-uniqueness or incompleteness of maximum likelihood solutions to the Group-Lasso functional. The prediction performance on a test set (correlation with the true labels $\rho = 0.66$) was practically identical to the results reported in [12] and in the original paper [44], which has been viewed as among the best methods for short motive modeling.

4.7 Summary

In this chapter, we extended our Bayesian grouped-variable selection framework to include generalized linear models other than standard linear regression. Having specified a general

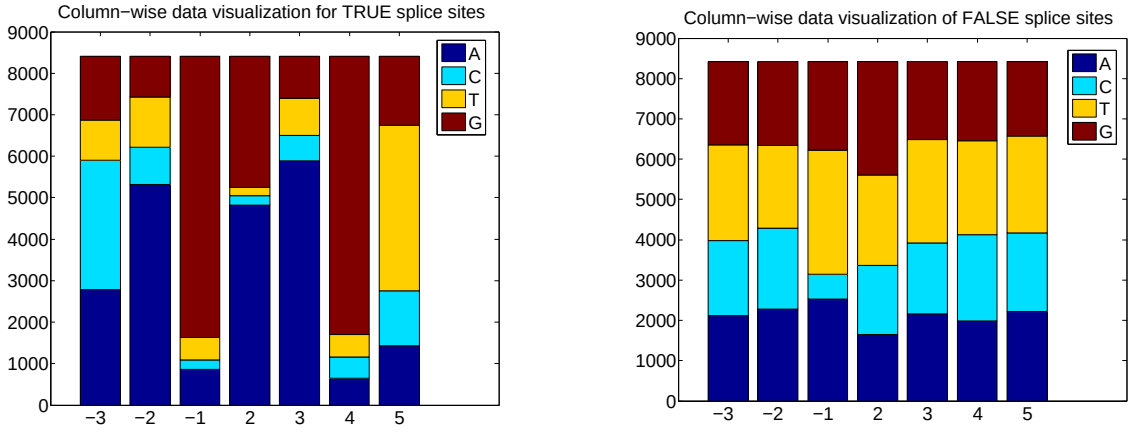


Figure 4.13.: Visualization of the MEMset data. **Left panel:** Distribution of A,C,T,G in the 7 window positions for the dataset with TRUE splice sites.**Right panel:** Distribution of A,C,T,G in the 7 window positions for the dataset with FALSE splice sites.

approach to the extension, we focused on two widely used GLMs, i.e. the Poisson and the binomial models which are frequently used for analyzing count data in contingency tables and classification problems respectively. We observed the ease with which the extension to GLMs was achieved with minimal changes to the Gibbs sampling algorithm implemented in the previous chapter. For the Poisson model applied to contingency tables, we achieved a further optimization of the Gibbs sampling algorithm which made the inference practical for large scale problems.

We also looked at extending the analysis to higher-order interactions. This lead to an alternate interpretation of the sparse variable selection problem wherein it was interpreted as a sparse hypergraph learning problem where the nodes represent the predictor variables. Such an interpretation was found to be useful when compared to other network models like Bayesian networks which usually do not model higher-order edges. These ideas were then illustrated through real-world biological applications dealing with bio-marker detection for breast cancer and splice site detection in human DNA sequences.

In the next chapter, we look at an example of a non-standard generalized linear model which is motivated by the domain of survival analysis, namely the Weibull model and apply our variable selection framework on it. Motivated by a biological application of analyzing the factors affecting survival patterns in sub-groups, we also discuss the extension of the model to clustering via a mixture-of-experts model, where the goal is to simultaneously learn clusters in data based on survival patterns, along with identifying significant features in each cluster which characterize this pattern.

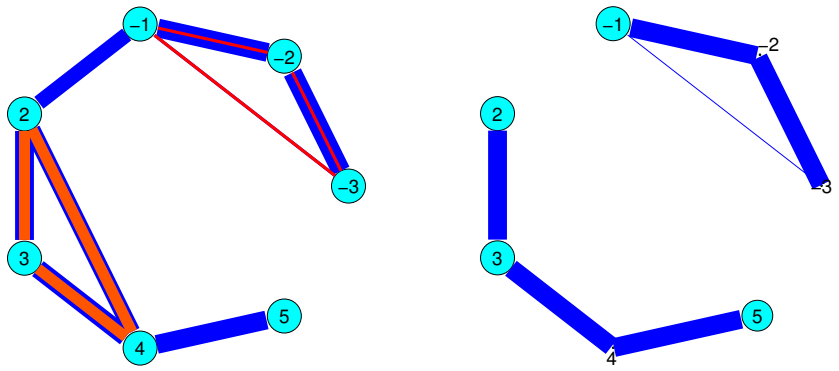


Figure 4.14.: Results of the interaction patterns for true and false splice sites.The thickness of the lines indicate the significance of the interactions. **Left panel:** Interaction patterns of the TRUE splice sites.**Right panel:** Interaction patterns of FALSE splice sites.

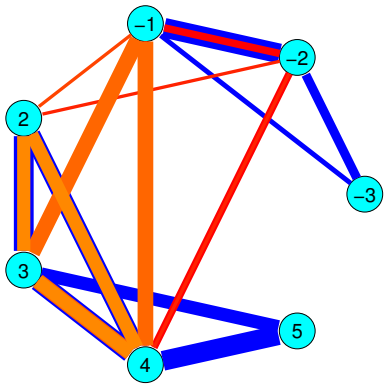


Figure 4.15.: Results of the interaction patterns for the classification between true and false splice sites. The thickness of the lines indicate the significance of the interactions.

5

Mixture-of-Experts Model for Survival Analysis

5.1 Survival Analysis

In this chapter, our focus will be on the application domain of **survival analysis**. We will see how certain problems in survival analysis can be analyzed by suitably extending our Bayesian framework for grouped-variable selection. Our motivation is primarily from a real-world biological application which deals with the analysis of survival patterns in patients with breast cancer.

Survival analysis is a branch of statistics dealing with the analysis of time-to-failure data and is applicable to a variety of domains like biology, engineering, economics etc. More generally, it is the analysis of time-to-event data where an event could signify death, failure etc. Particularly in the context of disease studies, it is a powerful tool for identifying and analyzing the differences in survival patterns between various patient sub-groups. Another useful analysis in this respect can be to identify some key patient attributes which might potentially contribute to the difference in survival patterns observed between various sub-groups. However, the sub-groups are not always identified in advance and it can be interesting from an application point of view to detect possible sub-groups, within a patient group, based on the differences in their survival patterns.

In this chapter, we will first introduce some basic concepts related to survival analysis which include the terminology and some popular models for analyzing the effect of predictor variables on the response variable which in this case is the survival time. We will then introduce the type of problems that we are interested to address within the realm of survival analysis. We then show how we can reuse our Bayesian framework for grouped variable selection for solving these problems with suitable model extensions. On the algorithm side, we show how the MCMC inference algorithm is easily extended to incorporate the resulting complicated hierarchical model. This will be done in two stages. In the first stage, we will tackle a simpler problem by assuming the data to be homogeneous in nature. In the second stage, we will use the simple case to build a more generic model which deals with the analysis of heterogeneous data via clustering.

5.2 Survival Regression

In this section, we first introduce some basic terminology in survival analysis which will help us explain the extensions to our Bayesian framework for grouped variable selection in subsequent sections. We will initially assume that the data is generated based on a homogeneous group of patients who share a common survival pattern and also a common effect of the predictor variables on the survival pattern.

Survival analysis deals with the analysis of time-to-event data and the observations are in the form of survival time. A parametric approach to survival analysis involves the estimation of parameters of a probability density function which models survival time, where time is represented by a continuous non-negative random variable T i.e. $T \in [0, \infty)$. Let $f(t)$ and $F(t)$ denote the probability density function and cumulative density function of T respectively:

$$F(t) = P(T \leq t) = \int_0^t f(u)du. \tag{5.1}$$

Further, a **survival function** S is defined based on the cumulative distribution function of T as follows:

$$S(t) = P(T > t) = 1 - F(t) = 1 - \int_0^t p(u)du, \tag{5.2}$$

which models the probability of an individual surviving up to time t or an event not occurring up to time t.
The **hazard function** $h(t)$ measures the instantaneous rate of failure at time t provided the individual survives till time t and is defined as follows:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} = \frac{f(t)}{S(t)}. \tag{5.3}$$

Based on the eq. 5.3, we can also derive the following expression:

$$h(t) = -\frac{d}{dt} \log(S(t)). \tag{5.4}$$

To model the effect of predictor variables on the survival time, there are two popular approaches, i.e. the proportional hazards model and the accelerated failure time model.

5.2.1. Effect of Predictor Variables on Survival

The above modeling of time is further extended by considering the effect of predictor variables $\boldsymbol{x}$ on time t via a regression component. We will discuss two popular models in this context.

Proportional Hazards Model. One way to model the effect of predictor variables is via the effect on the hazard function which can depend on time t as well as a set of variables $\boldsymbol{x}$. A widely used model which models the affect of the predictor variables in a multiplicative

manner is the proportional hazards model introduced in [45] which defines the hazard function as:

$$h(t|\boldsymbol{x}) = h_0(t)\exp(f(\boldsymbol{x}, \boldsymbol{\beta})), \tag{5.5}$$

where $h_0(t)$ is the baseline hazard function, $\boldsymbol{x}$ is the vector of predictor variables and $\boldsymbol{\beta}$ is a vector of regression coefficients. The popular choice for f is a linear function:

$$f(\boldsymbol{x}, \boldsymbol{\beta}) = \eta = \boldsymbol{x}^T\boldsymbol{\beta}. \tag{5.6}$$

Accelerated Failure Time Model. This is an alternative choice to the proportional hazards model for modeling the effects of predictor variables on survival time. Instead of using the hazard function, the effects are modeled directly as a multiplicative factor on the observed time, which results in rescaling the original time:

$$t = t_0\exp(f(\boldsymbol{x}^t\boldsymbol{\beta})), \tag{5.7}$$

where t_0 is the baseline survival time and, as before, the usual choice for f is the linear function $\boldsymbol{x}^t\boldsymbol{\beta}$. For a linear function, the hazard function for this model results in:

$$h(t) = h_0(t_0\exp(\boldsymbol{x}^t\boldsymbol{\beta}))\exp(\boldsymbol{x}^t\boldsymbol{\beta}). \tag{5.8}$$

As shown in the previous chapter, we also consider higher-order interactions upto second-order instead of modeling just the main effects (individual features).

Censoring. Another aspect of survival analysis is the presence of missing values in data. Data is often right-censored, which means that for individuals whose exact survival time t is not known, t indicates that the survival was atleast till that point in time. For the experiments in this document, we will consider only right-censored data. The censoring for a data point is indicated by a random variable δ, where $\delta = 0$ indicates that the data point is censored. For a right-censored proportional hazards model, the likelihood function is defined as:

$$p(\{t_i\}_{i=1}^n|\boldsymbol{\beta}, X) = \prod_{i=1}^{n}[h_0(t_i)\exp(\boldsymbol{x}_i^t\boldsymbol{\beta})]^{\delta_i}(S_0(t_i)^{\exp(\boldsymbol{x}_i^t\boldsymbol{\beta})}), \tag{5.9}$$

where X denotes the design matrix with rows representing single observations (censored and uncensored) and S_0 is the baseline survival function and n is the number of observations. Additionally, more flexibility is added to the model by adding a random effect to the linear function f:

$$\eta_i = \boldsymbol{x}_i^t\boldsymbol{\beta} + \epsilon_i, \qquad \text{where} \quad \epsilon_i \sim N(0, \sigma^2) \qquad \forall\ i = 1 \ldots n. \tag{5.10}$$

We will show how this also serves an alternate purpose of simplifying the Gibbs sampling algorithm.

5.2.2. Weibull Distribution

Although there are various choices of distributions for modeling T, like the gamma, exponential and log-normal distributions, our choice of distribution for modeling time is the Weibull distribution defined as:

$$p(t|\alpha_w, \lambda_w) = \alpha_w \frac{1}{\lambda_w} t^{\alpha_w - 1} \exp\left(-\frac{1}{\lambda_w} t^{\alpha_w}\right), \tag{5.11}$$

where $t > 0, \alpha_w > 0, \lambda_w > 0$ and α_w and λ_w represent the shape and scale parameters respectively. For $\alpha_w = 1$, it represents the exponential distribution. The corresponding hazard and survival functions are:

$$\begin{aligned} h(t) &= \alpha_w \frac{1}{\lambda_w} t^{\alpha_w - 1} \\ S(t) &= \exp\left(-\frac{1}{\lambda_w} t^{\alpha_w}\right). \end{aligned} \tag{5.12}$$

It is one of the most widely used distributions for survival analysis due to a couple of reasons. Firstly, it is a very flexible distribution since it can model a variety of survival functions and hazard rates. Apart from flexibility, it is also the only distribution for which both the accelerated failure time model and the proportionality hazards model coincide ([46]). This means that if we start with two hazard functions and multiply one with a relative risk and for the other time is stretched, then both effects can be represented equivalently with a Weibull distribution (see Appendix B for a standard derivation based on [47]). Assuming right-censored data, the likelihood is written as:

$$p(\boldsymbol{t}|\alpha_w, \lambda_w) = \prod_{i=1}^{n} \left(\frac{\alpha_w}{\lambda_w} t_i^{\alpha_w - 1}\right)^{\delta_i} \exp\left(-\frac{1}{\lambda_w} t_i^{\alpha_w}\right), \tag{5.13}$$

where n is the number of observations, $\delta_i = 0$ when the i^{th} observation is censored and $\delta_i = 1$ otherwise and $\boldsymbol{t} = (t_1, t_2, \cdots, t_n)$. Further, to model the effect of $\boldsymbol{x}$ on the distribution over time, we apply the proportional hazards model. Based on eq. (5.9) the modified likelihood including the effect of $\boldsymbol{x}$ is as follows:

$$p(\boldsymbol{t}|\boldsymbol{\eta}, \alpha_w, \lambda_w) = \prod_{i=1}^{n} \left[\frac{\alpha_w}{\lambda_w} t_i^{\alpha_w - 1} \exp(\eta_i)\right]^{\delta_i} \exp\left(-\frac{1}{\lambda_w} t_i^{\alpha_w} \exp(\eta_i)\right), \tag{5.14}$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_n)$ and $\eta_i \sim N(\eta_i|\boldsymbol{x}^t\boldsymbol{\beta}, \sigma^2)$.

We note that most parts of the Weibull regression model described so far resemble a random-intercept model which is a type of GLM. But it is not strictly a GLM since the Weibull distribution lacks fixed-length sufficient statistics and is not considered, in a strict sense, to be part of the exponential family of distributions unless the shape parameter is known. In order to provide a full Bayesian treatment of the model, we can define suitable priors for the parameters of the model, namely σ, $\boldsymbol{\beta}$, α_w, and λ_w.

Contrast Coding. In this chapter, we will focus on predictor variables which are categorical in nature. Categorical data is frequently encountered in biological applications and since our primary motivation for survival analysis stems from biology, we briefly discuss the construction of the design matrix for categorical variables. When $\boldsymbol{x}$ is categorical in nature, a dummy coding procedure is applied in order to obtain a transformed design matrix X for inference. As described in the previous chapter, this results in replacing each categorical variable with a group of dummy variables. Apart from single variables (interactions of order zero), the design matrix also consists of higher-order terms i.e. 1st order (pairwise interactions) and 2nd order (triplet interactions) terms. An example of an observation matrix consisting of two variable with three categories each along with a first-order interaction transformed using dummy coding is shown in Figure 5.1.

We choose polynomial contrast codes since they are suited for ordered categorical variables and avoid over-parametrization by representing a K-level variable with $K-1$ columns (see Figure 5.1 bottom). This results in representing each categorical variable as a group of contrast-coded variables. Hence, to create the full design matrix, first the levels are contrast-coded which gives us the codes for respective levels (see Figure 5.1 bottom-right) and then each observation is recoded (for main effects and higher-order interactions) using these codes as reference.

Mixture Models. Before moving on to the problems that we will analyze in survival analysis, we will introduce concepts and terminology related to mixture models which will be used in the rest of this chapter. Mixture models ([32]) are used in the context of identifying groups or clusters of data points in multidimensional space. Given a set of observations $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$, the goal is to partition the data into k clusters where we will initially assume k to be given. From a modeling perspective, the data is assumed to be generated from mixture of distributions. We will explain further details by taking the example of a mixture model consisting of a mixture of normal distributions.

Assuming that data is generated from a mixture of k normal distributions, we define the conditional distribution of a data point, given the particular cluster it belongs to, as a normal distribution:

$$\boldsymbol{x}_i | c_j \sim N(\boldsymbol{x}_i | \boldsymbol{\mu}_{c_j}, \sigma^2_{c_j}), \tag{5.15}$$

where $c_j \in \{1, 2, \cdots, k\}$ is an index to the jth cluster representing a normal distribution in this case. Further we define weights $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_k)$ given to each cluster which sum up to one:

$$p(c_j) = \pi_j \qquad \text{where} \sum_{j=1}^{k} \pi_j = 1. \tag{5.16}$$

These weights are known as mixing proportions or mixture components. Hence, the marginal likelihood for one data point is written as:

$$p(\boldsymbol{x}_i | \bullet) = \sum_{j=1}^{k} \pi_j N(\boldsymbol{x}_i | \boldsymbol{\mu}_{c_j}, \sigma^2_{c_j}). \tag{5.17}$$

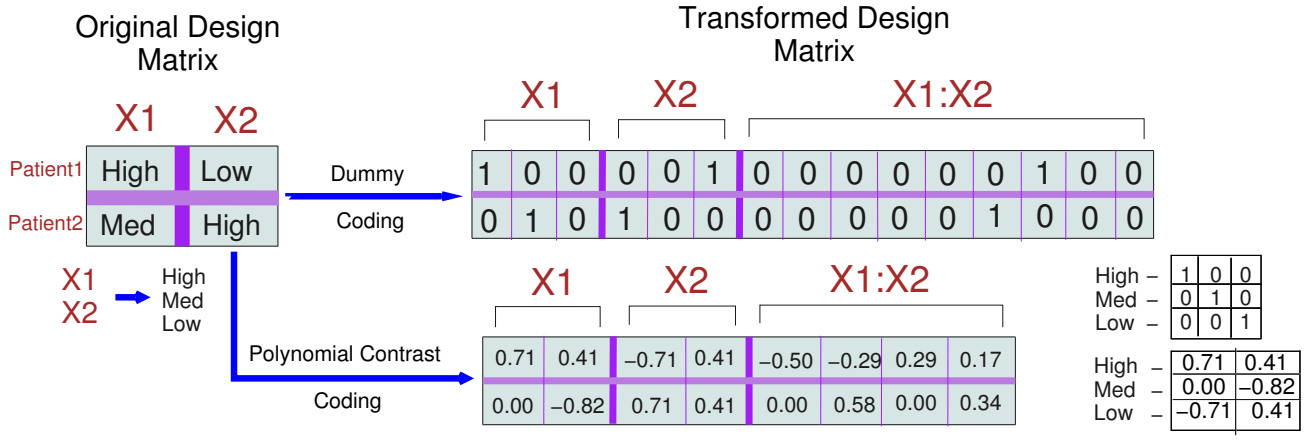


Figure 5.1.: **Dummy coding illustration:** On the top-left, we have categorical observations for 2 patients for whom 2 biomarkers ($X1$ and $X2$) are measured for expression values. Each categorical variable can have three possible values (high, med and low). The top-right side shows the transformed design matrix after a standard dummy coding procedure has been applied. The resulting design matrix represents each variable as a group of dummy-variables. Hence identifying key features from the original matrix is translated to the problem of identifying key *groups* of dummy variables. The bottom-right shows the transformed matrix after using a polynomial contrast coding procedure. The resulting contrast-coded matrix uses $(K-1)^{order+1}$ columns for an interaction as opposed to $(K)^{order+1}$ columns in a a dummy-coded matrix where K is the number of categories for a variable and *order* denotes the order number of the interaction (zeroth, first, second etc).

Further priors can be placed on all the unknowns which include the cluster parameters and the mixing proportions. The conjugate prior on the mixing proportions is the Dirichlet distribution. The full model is described as:

$$\begin{aligned} \boldsymbol{x}_i \mid c_i,\ \boldsymbol{\phi}_{c_i} &\sim F(\boldsymbol{\phi}_{c_i}) \\ c_i \mid \boldsymbol{\pi} &\sim G(\pi_1, \pi_2, \cdots, \pi_k) \\ G &\sim \text{Dir}\left(\left(\underbrace{\frac{1}{k}, ..., \frac{1}{k}}_{k \text{ times}}\right), \alpha_d\right), \end{aligned} \tag{5.18}$$

where $\boldsymbol{\phi}_c$ denotes the parameters of a cluster with G_0 as the prior distribution, G denotes a discrete distribution and "Dir" denotes the Dirichlet distribution as defined in Appendix A.

This example can be easily generalized to other distributions by replacing the normal distribution with a different one. For a fixed k, the inference in mixture models involves the learning of the parameters of each cluster and also the assignment of data points to clusters i.e. which data point belongs to which cluster. For inference, MCMC sampling

algorithms can be used for sampling both the parameters and the assignment vectors. Then based on the samples accumulated, the assignment of data points to clusters can be estimated.

The version of mixture models discussed so far are finite mixture models in the sense that the maximum number of clusters is predefined. But in most applications, the number of clusters is not known in advance. For such cases, a modified version called the infinite mixture model ([48]) can be used since it does not fix the number of clusters in advance. In this case the prior over the mixing proportions is defined to be a Dirichlet Process (DP). The Dirichlet process is a distribution on distributions i.e. a particular sample from a DP is also a probability distribution from which samples can be drawn. The draws from a DP are discrete hence making it a useful prior for clustering purposes. The full model is written as:

$$\begin{aligned} \boldsymbol{x}_i \mid c_i,\ \boldsymbol{\phi}_{c_i} &\sim F(\boldsymbol{\phi}_{c_i}) \\ c_i \mid \boldsymbol{\pi} &\sim G \\ G &\sim DP(G_0, \alpha_d), \end{aligned} \tag{5.19}$$

where DP denotes a Dirichlet Process and G_0 is the base distribution.

Mixture-of-Experts. A mixture-of-experts (MOE) model, as proposed in [49] (see Figure 5.2: left panel), uses a divide-and-conquer strategy to represent a complex clustering problem by localizing it over different regions in the feature space where features are represented by variable $\boldsymbol{x}$. The model assumes that data can be summarized by a set of localized mixture distributions where the mixing components are known as experts. Hence clusters or mixing components, represented by experts, are probability distributions over a variable, say $\boldsymbol{z}$, conditioned on the features $\boldsymbol{x}$. The distribution of $\boldsymbol{z}$ can be written based on a standard mixture model conditioned on $\boldsymbol{x}$ which models data that is generated based from a mixture of distributions:

$$p(\boldsymbol{z}|\boldsymbol{x}, \bullet) = \sum_{j=1}^{K} p(c_j|\boldsymbol{x}, \bullet) p(\boldsymbol{z}|\boldsymbol{x}, c_j, \bullet), \tag{5.20}$$

where $(\bullet)$ represents all the unknown parameters, c_j's are the mixture components and K is the number of clusters which is known in advance. The first term in the left hand side of eq. (5.20), i.e. $p(c_j|\boldsymbol{x}, \bullet)$, is the **gate function** which decides the weight given to the j^{th} expert based on $\boldsymbol{x}$. The weight denotes how well an expert explains the survival pattern for that region of the space over predictor variables. Using Bayes' rule, we can rewrite the model in the following way in order to resemble a standard mixture model, as shown in [50]):

$$p(\boldsymbol{z}|\boldsymbol{x}, \bullet) \propto \sum_{j=1}^{K} p(c_j) p(\boldsymbol{x}|c_j, \bullet) p(\boldsymbol{z}|\boldsymbol{x}, c_j, \bullet). \tag{5.21}$$

This representation allows us to visualize each mixture component as a joint distribution over $(\boldsymbol{x}, \boldsymbol{z})$. As in mixture models, a Dirichlet distribution can be used as a conjugate prior over the mixing proportions.

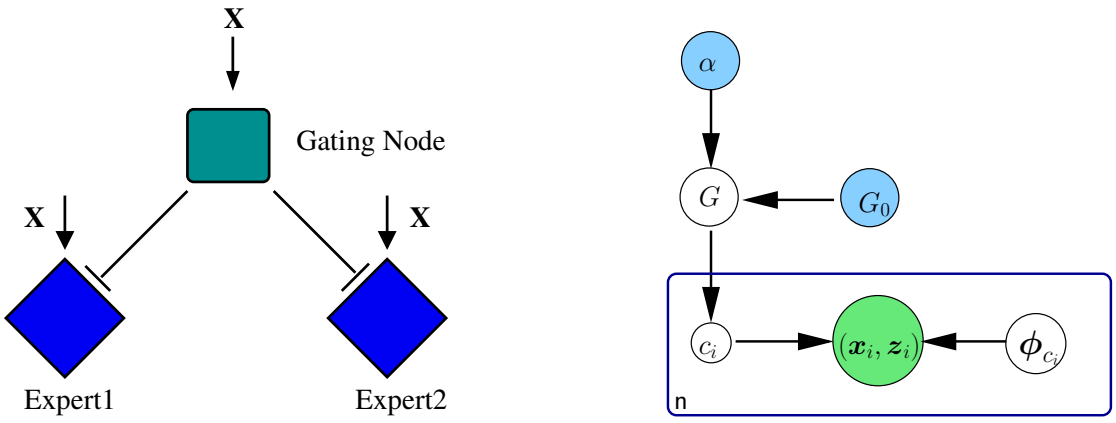


Figure 5.2.: **Left panel:** Mixture-of-experts model for two experts with a gating node representing the function that decides which of the two experts is chosen to make a prediction for $\boldsymbol{x}$ which is represented by $p(c_j|\boldsymbol{x}, \bullet)$ in eq. (5.20). **Right panel:** Infinite mixture of experts using a Dirichlet process prior G with parameters (α, G_0). The number of observations is denoted by n and the respective assignment variables by c_i. The observed variables $\boldsymbol{x}$ and $\boldsymbol{z}$ are represented in green with the priors collapsed to $\boldsymbol{\phi}_{c_i}$.

The above model was described for the case when the underlying number of clusters is fixed. A further enhancement to the finite mixture-of-experts model involves removing this limiting assumption as well. The model is extended to an infinite mixture-of-experts by replacing finite clusters with infinite clusters ([48]) and hence replacing the Dirichlet distribution by a Dirichlet process (DP) as prior for the mixing proportions, similar to [50]. The extension to an infinite mixture-of-experts model is described in a hierarchical manner as follows (see right panel of Figure 5.2):

$$\begin{aligned} (\boldsymbol{x}_i, \boldsymbol{z}_i) \mid c_i,\ \boldsymbol{\phi}_{c_i} &\sim F(\boldsymbol{\phi}_{c_i}) \\ c_i \mid \boldsymbol{\pi} &\sim G \\ G &\sim DP(G_0, \alpha_d), \end{aligned} \tag{5.22}$$

where DP denotes a Dirichlet process prior with base distribution G_0 and a concentration parameter α_d, c_i is the latent class to which an observation $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ belongs and $\boldsymbol{\phi}_c$ denotes the parameters which determine the distribution of class c. The effective number of clusters can be inferred from data by carrying out MCMC sampling with various choices of algorithms (see [51]).

5.2.3. A Unified Framework for Survival Analysis

In the past, the proportionality hazards model has been extended to a mixture model in order to find sub-groups in data with respect to survival time and to measure the effect of predictor variables within each sub-group. In this context, a *finite* mixture-of-experts

(MOE) model is defined in [52] by maximizing the partial likelihood for the regression coefficients and by using some heuristics to resolve the number of experts in the model. The mixture of experts model is described more in detail later in the chapter. A more recent attempt at this analysis, which is carried out in [53], uses a maximum likelihood approach to infer the parameters of the model and the Akaike information criterion (AIC) to determine the number of mixture components. A Bayesian version of the mixture model has been investigated in [54], which analyzes the model with respect to time but does not capture the effect of predictor variables. On the other hand, the work in [55] performs variable selection based on the predictor variables but ignores the clustering aspect of the modeling. Similarly, the infinite mixture model defined in [56] does not include a mixture of experts, hence assumes all the predictor variables to be generated from the same distribution and also uses a common shape parameter for modeling the Weibull distribution for each expert.

Based on the existing body of work, our goal is to unify the various important elements of survival analysis discussed in the models above into a Bayesian infinite mixture-of-experts (MOE) framework. We will use this framework to model survival time, while capturing the effect of predictor variables through variable selection and will also deal with an unknown number of mixing components. From the perspective of our Bayesian framework for grouped variable selection, we will build extensions to cater to the Weibull model using the proportionality hazards model. Secondly, we introduce a further extension to include clustering via a mixture-of-experts model in order to simultaneously learn the clusters existing in data based on survival time and also the significant predictor variables via the regression coefficients in the Weibull linear regression model. Similar to the previous chapter, the concept of grouped variable selection is applied to categorical variables which are commonly encountered in biological applications.

5.3 Survival Analysis with Variable Selection

Based on the goals laid out in the previous section, we will first start by looking at a single cluster model assuming the data to be homogeneous. Since our goal is variable selection in the context of survival analysis, we start with the likelihood model defined in eq. (5.14) and then define suitable priors on the parameters of the model, namely σ, $\boldsymbol{\beta}$, α_w, and λ_w.

Prior Distributions. As described in the previous section, the dummy variable coding procedure gives rise to groups of dummy-coded variables. This transformation of data leads to the task of selecting predictor variables on a group level, i.e. *grouped* dummy variables, where each group represents a single categorical predictor variable. We now apply our Bayesian framework for grouped-variable selection to the Weibull regression model in order to select significant interaction terms which include higher-order terms up to order two. This results in including the variables α, Λ and ρ in the model via the Bayesian grouped-variable selection framework.

The prior specification for $\boldsymbol{\beta}$ and σ^2 is the same as described for the GLM in the previous chapter. All other variables introduced in the model (α, Λ and ρ) also are treated in the

same manner. The predictor variables $\boldsymbol{x}$ are assumed to be generated from a normal distribution:

$$p(\boldsymbol{x}|\boldsymbol{\mu}_c, \sigma_c^2) = \mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \sigma_c^2). \tag{5.23}$$

For a full Bayesian treatment, we apply standard joint conjugate prior for $\boldsymbol{\mu}_c$ and σ_c^2 as follows:

$$\begin{aligned} \boldsymbol{\mu}_c|\sigma_c^2 &\sim \mathrm{N}(\boldsymbol{\mu}_c|\boldsymbol{\mu}_{0c}, \tau^{-1}I) \\ \sigma_c^2 &\sim \mathrm{InvGamma}(\sigma_c^2|e, h). \end{aligned} \tag{5.24}$$

Although the Weibull distribution lacks fixed-length sufficient statistics, we define a joint conjugate prior for the two parameters (α_w,λ_w), based on the work in [28]:

$$p(\alpha_w, \lambda_w|a, b, c, d) \propto \alpha_w^{a-1} \exp(-\alpha_w b)\lambda_w^{-c} \exp\left(-\frac{d^{\alpha_w}}{\lambda_w}\right), \tag{5.25}$$

where $a, b, c > 0$ and d allows us to deal with the lack of fixed-length sufficient statistics. The full model with all the variables is graphically described in Figure 5.3.

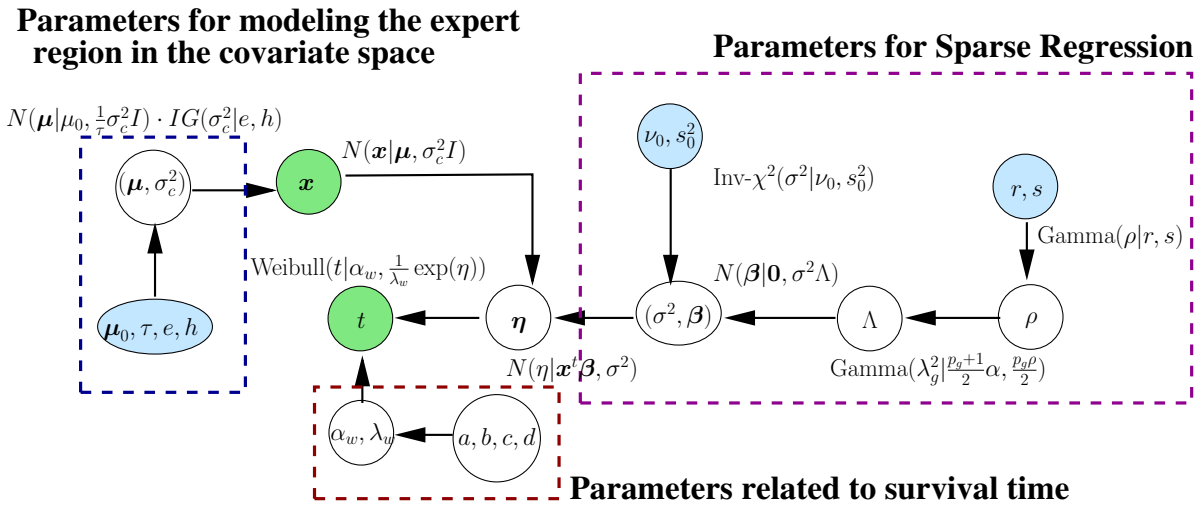


Figure 5.3.: The complete hierarchical model with the parameterization for a single cluster model. Depicted in blue are the hyperparameters for the respective distributions, like (r, s) for the Gamma prior on ρ. The observed variables $\boldsymbol{x}$ and t are shown in green. The part of the figure centered around t forms the core which defines the generalized linear model with a normally distributed random link between η and $\boldsymbol{x}$ and coefficients and priors for the Weibull distribution. The block on the right defines the hierarchy related to the sparse regression on the predictor variables via the hierarchical representation of the normal-gamma prior on the regression coefficients $\boldsymbol{\beta}$. Furthermore, the left block defines the variables for describing the distribution of the space over $\boldsymbol{x}$.

Posterior Conditional Distributions. For posterior inference, we again reuse our framework for Bayesian grouped variable selection. We extend Algorithm 1 which was used for grouped variable selection in least squares regression. The only change required is the augmentation of steps for sampling the variables $\boldsymbol{\mu}_c, \sigma_c^2, \alpha_w$, λ_w and $\boldsymbol{\eta}$. All the other posterior conditional distributions for $\boldsymbol{\beta}$, σ^2, Λ and ρ remain the same as in the least-squares regression case.

Due to conjugacy, the posterior conditional distribution of $(\boldsymbol{\mu}_c, \sigma_c^2)$ is again split into an normal and inverse-gamma distribution:

$$\begin{aligned}\mu_c|\sigma_c^2, \bullet &\sim \mathrm{N}\left(\mu_c|\frac{\sum_{i=1}^n \boldsymbol{x}_i + \tau\mu_{0c}}{\tau+n}, \tau^{-1}\sigma_c^2 I\right)\\ \sigma_c^2|\bullet &\sim \mathrm{IG}\left(\sigma_c^2|k+n, k\theta + \sum_{i=1}^n(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^t(\boldsymbol{x}_i-\bar{\boldsymbol{x}}) + \frac{\tau n}{\tau+n}(\boldsymbol{\mu}_{0c}-\bar{\boldsymbol{x}})^t(\boldsymbol{\mu}_{0c}-\bar{\boldsymbol{x}})\right).\end{aligned} \tag{5.26}$$

The posterior conditional distribution of η_i is difficult to sample from since it is not of standard form:

$$p(\eta_i|\bullet) = \left[\frac{\alpha_w}{\lambda_w} t_i^{\alpha_w - 1}\exp(\eta_i)\right]^{\delta_i} \exp\left(-\frac{1}{\lambda_w} t_i^{\alpha_w}\exp(\eta_i)\right) N(\eta_i|\boldsymbol{x}_i^t\boldsymbol{\beta}, \sigma^2). \tag{5.27}$$

However, since the posterior conditional distribution is log-concave, we propose the use of Laplace approximation (see [31]), which approximates the posterior conditional distribution to a normal distribution and simplifies sampling considerably. The posterior conditional distribution for η_i is approximated to a normal distribution $N(\eta_i|\eta_{i0}, c_0)$ where η_{i0} is the solution to the equation:

$$\delta_i - \frac{t_i^{\alpha_w}}{\lambda_w}\exp(\eta_i) + \frac{1}{\sigma^2}(\boldsymbol{x}_i^t\boldsymbol{\beta} - \eta_i) = 0. \tag{5.28}$$

With this value, the value of c_0 is calculated by evaluating:

$$c_0 = \frac{t_i^{\alpha_w}}{\lambda_w}\exp(\eta_{i0}) + \frac{1}{\sigma^2}. \tag{5.29}$$

For the Weibull parameters α_w and λ_w, sampling based on their individual posteriors conditioned on each other is avoided, since this results in a slow mixing of the Markov chain due to a high correlation between samples from the two conditionals. To overcome this issue, the posterior conditional of (α_w, λ_w) is split up into the conditional of λ_w given α_w which results in an inverse-gamma distribution,

$$p(\lambda_w|\alpha_w, \bullet) \propto \mathrm{IG}\left(c + h_d - 1, d^{\alpha_w} + \textstyle\sum_{i=1}^n t_i^{\alpha_w}\exp(\eta_i)\right), \tag{5.30}$$

where h_d is the number of deaths (number of data points for which $\delta_i = 1$) and the marginal

of α_w which is derived based on the work in [28]:

$$p(\alpha_w|\bullet) \propto \frac{\alpha_w^{a+h_d-1} \exp(-\alpha_w(b - \log(P_{hd})))}{(d^{\alpha_w} + \sum_{i=1}^{n} t_i^{\alpha_w} \exp(\eta_i))^{c+h_d-1}}, \tag{5.31}$$

where P_{hd} is the product of t_i's for which $\delta_i = 1$ and $(\bullet)$ represents all the unknown parameters. However, since this marginal distribution is non-standard, we discretize α_w to simplify sampling. The complete MCMC algorithm is given in Algorithm 3. Based on

Algorithm 3 Gibbs sampling for survival analysis

1: **Input:** n observations $D = (\boldsymbol{x}_i, t_i)$.
2: **Initialize:** Parameters $\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \rho, \Lambda, \alpha$.
3: Draw samples of the unknowns from the posterior joint distribution $p(\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \rho, \Lambda, \alpha|, D)$ by drawing from the conditionals.
4: **for** $m = 1$ to BayesIter **do**
5: Sample $\boldsymbol{\mu}_c, \sigma_c^2|\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\beta}, \sigma^2, \alpha, \Lambda, D$ - from a normal-inverse-gamma distribution given in eqn. 5.26.
6: Sample $\boldsymbol{\eta}$ $|\alpha_w, \lambda_w, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \alpha, \Lambda, D$ -based on a Laplace approximation of the distribution given in eqn. 5.27.
7: Sample λ_w $|\alpha_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \alpha, \Lambda, D$ - from the inverse-gamma distribution given in eqn. 5.30.
8: Sample α_w $|\lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \alpha, \Lambda, D$ - from a discretized version of the distribution given in eqn. 5.31.
9: Sample α $|\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \Lambda, D$ - from a discretized version of the distribution given in eqn. 3.19.
10: Sample ρ $|\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \Lambda, \alpha, D$ - from a Gamma distribution given in eqn. 3.18.
11: Sample Λ $|\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \boldsymbol{\beta}, \sigma^2, \rho, \alpha, D$ - from a generalized inverse gaussian distribution given in eqn. 3.17.
12: Sample $\boldsymbol{\beta}, \sigma^2$ $|\alpha_w, \lambda_w, \boldsymbol{\eta}, \boldsymbol{\mu}_c, \sigma_c^2, \rho, \Lambda, \alpha, D$ - σ^2 is sampled from an inverse-gamma distribution (eqn. 4.10) and $\boldsymbol{\beta}$ conditioned on σ^2 from a multivariate Normal distribution (eqn. 3.15).
13: **end for**

these posterior sampling steps, there is an issue encountered with the sampling of λ_w and $\boldsymbol{\beta}$. Since both affect the scale of the Weibull distribution, there can be a competing effect of parameter estimation while sampling for these two parameters. To avoid this problem, we incorporated λ_w in the intercept term of the regression component $\boldsymbol{x}^t\boldsymbol{\beta}$.

5.4 Identifying Clusters of Survival Patterns

Till now we have assumed that the data is generated from one homogeneous group and hence all the analysis was done with respect to this one group. But this may not always be

the case and the data may represent a group which is a combination of different heterogeneous sub-groups where each sub-group exhibits a possibly different survival pattern. In such a case, it is important to identify these sub-groups in data and also simultaneously identify the significant predictor variables which influence the survival time distribution within each sub-group. This particular model is very appealing especially in many biological applications where there might be an interest in identifying sub-groups in patients. Identifying these sub-groups may potentially shed some light on the underlying biological problem being studied based on the differences between these groups. We perform our analysis using a **mixture-of-experts** (MOE) model which we first describe in the context of predefined number of clusters. To this, we add the component of sparse survival regression which we have described so far. The extended model then unifies the two tasks of cluster identification and variable selection. Since the number of clusters existing in data is not known in advance, we enhance this model further by using the infinite mixture-of-experts model which does not require the number of clusters to be specified in advance.

Finite Mixture of Experts. Initially, we started with the assumption that the data was generated from one homogeneous group. We further enhance this idea by removing this assumption and model data which is potentially generated from multiple, but a *known* number of sub-groups/clusters in data. In order to model the clusters in terms of the combined effects of the predictor variables $\boldsymbol{x}$ and survival time t, we use an MOE model as described earlier. In this case, the clusters or mixing components, represented by experts, are probability distributions over time t conditioned on $\boldsymbol{x}$. Based on eq. (5.21), the distribution of $(\boldsymbol{x}, t)$ can be written as:

$$p(t|\boldsymbol{x}, \bullet) \propto \sum_{j=1}^{K} p(c_j) p(\boldsymbol{x}|c_j, \bullet) p(t|\boldsymbol{x}, c_j, \bullet). \tag{5.32}$$

The distribution over $\boldsymbol{x}$ is modeled as a normal distribution $N(\boldsymbol{x}|\boldsymbol{\mu}, \sigma_c^2 I)$ as show in Figure 5.3. The standard joint conjugate prior of normal-Inv-χ^2 distribution is applied to the parameters $(\boldsymbol{\mu}, \sigma_c^2)$. In order to combine clustering with sparse variable selection, we attach the component of sparse survival regression to this model. This is done by defining the distribution of $p(t|\bullet)$ based on eq. (5.14). The rest of the parameters and their respective priors follow from the previous section.

The resulting posterior conditional distributions for the two new variables in the model $(\boldsymbol{\mu}_c, \sigma_c^2)$, are also of standard form and hence can be easily incorporated into the Gibbs sampling algorithm for sparse survival regression. To complete the Bayesian picture, we also apply a suitable prior to the mixing proportions $\boldsymbol{c} = (c_1, c_2, \cdots, c_K)$. In a finite MOE model, a Dirichlet distribution is a standard conjugate prior to the mixing proportions:

$$p(\boldsymbol{c}|G, \alpha_d) \sim \text{Dir}(\boldsymbol{G}, \alpha_d) = \frac{1}{B(\alpha_d)} \prod_{i=1}^{K} c_i^{g_i \alpha_d - 1}, \tag{5.33}$$

where K is the number of clusters, $\boldsymbol{G}$ is a vector $\{g_1, g_2, \cdots, g_K\}$ of K probability values summing up to 1 and $B(\cdot)$ is the beta function:

$$\mathrm{B}(\alpha_d) = \frac{\prod_{i=1}^{K} \Gamma(g_i \alpha_d)}{\Gamma\left(\sum_{i=1}^{K} g_i \alpha_d\right)}. \tag{5.34}$$

The posterior sampling for the assignment vectors and the mixing proportions can be done in a straightforward way based on the work in [57]. All other parameters follow our framework for Bayesian grouped-variable selection discussed in earlier chapters.

Infinite Mixture of Experts. We now add the final enhancement to the mixture-of-experts model by removing the assumption of knowing the number of clusters in advance. The model is extended to an infinite mixture-of-experts by replacing the Dirichlet distribution by a Dirichlet process (DP) as prior for the mixing proportions, similar to [50]. The extension to an infinite mixture-of-experts model for survival analysis is described as follows:

$$\begin{aligned} (\boldsymbol{x}_i, t_i) \mid c_i,\ \boldsymbol{\phi}_{c_i} &\sim F(\boldsymbol{\phi}_{c_i}) \\ c_i \mid \boldsymbol{\pi} &\sim G \\ G &\sim DP(G_0, \alpha_d), \end{aligned} \tag{5.35}$$

where all the variables are defined as before in eq. (5.22).

Algorithm 4 Blocked Gibbs Sampling for a Truncated Dirichlet process

1: **Input:** n observations $D = (\boldsymbol{x}_i, t_i)$.
2: **Initialize:** c_i = cluster assignments and parameters $\boldsymbol{\phi}_{c_i}$.
3: Draw from the posterior of the joint distribution $p(\boldsymbol{\pi}, \Phi^*, \boldsymbol{c}|D)$ by drawing from the conditionals.
4: **while** NotCoverged **do**
5: Sample $\Phi^* \mid \boldsymbol{\pi}, \boldsymbol{c}, D$ - This is carried out individually for each parameter in the model conditioned on the rest of the variables using Algorithm 3.
6: Sample $\boldsymbol{c} \mid \Phi^*, \boldsymbol{\pi}, D$ - For $i = 1, ..., N$, draw values
$$p(c_i|\boldsymbol{\pi}, \Phi^*, D) \sim p(c_i|\boldsymbol{\pi}) p(x_i, t_i|\phi_{c_i}), \quad c_i = 1, ..., M.$$
7: Sample $\boldsymbol{\pi} \mid \Phi^*, \boldsymbol{c}, D$ - The mixing proportions are drawn based on the posterior $p(\boldsymbol{\pi}|\alpha_d) p(\boldsymbol{c}|\boldsymbol{\pi})$.
8: **end while**

MCMC Sampling for Inference and Parameter Estimation. The inference of the infinite-mixture-of-experts model is carried out by MCMC sampling of the posterior distribution over the unknowns. Although there exist non-conjugate versions of the Dirichlet process algorithms (as given in [51]) which can be applied for inference, for practical reasons, we use a truncated version of the Dirichlet process called the Dirichlet-Multinomial allocation model [57], by specifying an upper bound on maximum number of clusters based

on the prior knowledge of the particular application. It serves as a good approximation to the DP measure and results in a finite-sum random probability measure which is computationally easy to implement. More specifically, we carry out a blocked-Gibbs sampling on a truncated Dirichlet process (see Algorithm 4 for details). After initializing all the parameters, the sampling algorithm is executed till an indication of convergence. The indication of convergence may be determined based on the length-control diagnosis explained in [38] or fixed to a number of iterations based on studying the trace plots of the sampling process in simulations.

5.5 Experiments

Simulations. In order to demonstrate the effectiveness of the model, experiments were carried out on simulated data. The first experiment shows the capability of the model to correctly identify two sub-groups in data along with identifying the key explanatory factors in both groups. The dataset of size 150 was generated from two equally proportioned clusters with $(5, 5)$ and $(1, 1)$ being the shape and scale parameters for the Weibull distribution for each cluster. The features consisted of 7 variables with expansion up to 2nd order interactions (63 terms). For the first cluster, the significant factors included main effects x_1, x_3 and x_4, all first order interactions with these three variables i.e. $(x_1 : x_3)$, $(x_1 : x_4)$, $(x_3 : x_4)$ and a second order interaction $(x_1 : x_3 : x_4)$. Similarly, for the second cluster, the significant factors included main effects x_2, x_6 and x_7, all first order interactions with these three variables i.e. $(x_2 : x_6)$, $(x_2 : x_7)$, $(x_6 : x_7)$ and a second order interaction $(x_2 : x_6 : x_7)$. Significance was achieved by assigning $\boldsymbol{\beta}$ values of $(3, 3, 3, 3, 3, 3, 3)$ and $(3, 3, 3, 3, 3, 3, 3)$ to the specific factors in the respective clusters and the rest of the $\boldsymbol{\beta}$ coefficients to zero. The predictor variables themselves were sampled from a normal distribution with means $(0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3)$ and $(0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7)$ for each cluster respectively. The Gibbs sampling process was executed for $50,000$ iterations and the burn-in was observed to be very early (in the first ≈ 100 iterations). Both the clusters were detected and all the true significant factors for both clusters were identified successfully. See Figure 5.4 for details.

In the second experiment, we compare our mixture-of-experts model to a global single cluster model in order to justify the need for a mixture model. The training data generated in the first experiment was used again for learning the parameters of a single-cluster model. In order to compare the two models, a separate test set (of size 500) was generated additionally to evaluate the performance of both models by comparing the log-likelihood of all the test points based on the parameters learned by both models. The per-point comparison is shown in Figure 5.5 which indicates the improvement achieved by using a MOE model.

We also performed a standard Kruskal-Wallis rank test which also ranks the MOE model higher than the single cluster model (see Figure 5.5 left panel). Apart from the quantitative evaluation, we also see in terms of identifying the significant factors (see Figure 5.5 right panel), that the single cluster model does poorly, both in recognizing the true factors and in terms of false positives. This can be explained based on the fact that

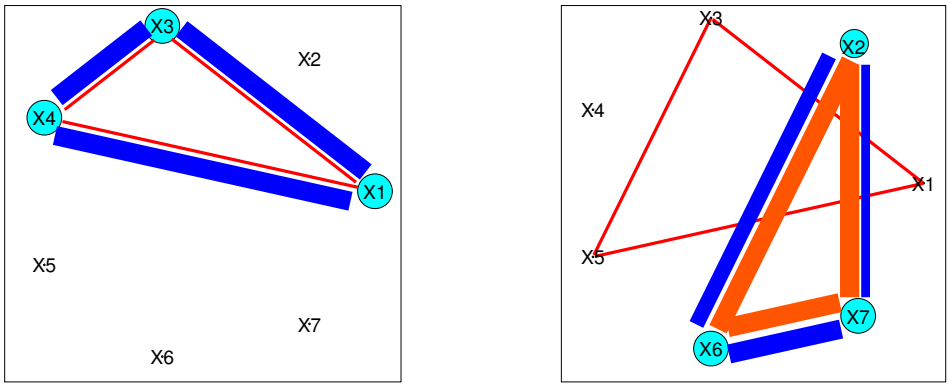


Figure 5.4.: Results for simulated data which was generated for 2 clusters with 7 categorical variables having interaction terms up to second order. In all interaction graphs, the light-blue circles represent the main effects, the blue lines represent 1st-order pairs and the reddish triangular lines indicate 2nd-order triplet interactions. In each case the size of the circle or the width of the lines indicates the estimated significance of the interaction: i.e. For example on the right cluster, more than 90% of the posterior samples for variable 2 have a positive sign. Based on the results of the inference process, we observe that all the key features have been correctly identified.

in a single cluster model, the model has to assume a common baseline model for both clusters. Then, in order to adjust for the real survival patterns, it can only achieve the same effect by making suitable adjustments to the regression component. In doing so, the model compromises in terms of the identification of significant factors from data. As a result, we see that the MOE model performs much better than a one-cluster model, hence justifying the need for a cluster-based model.

Application to Breast Cancer. The dataset consists of measured intensity levels obtained from tissue microarrays of the following markers: karyopherin-alpha-2 (KPNA2), nuclear staining for p53, the anti-cytokeratin CK5/6, the fibrous structural protein Collagen-VI, the inter-α-trypsin inhibitor ITIH5, the estrogen receptor (ER) and the human epidermal growth factor receptor HER2. From these categorical variables we constructed a design matrix which includes all dummy-coded interactions up to the second order.

Despite the fact that this dataset is one of the biggest of its kind, the rather low number of samples (270 patients) remains the main challenge in these scenarios. A further difficulty is the large number of censored patients (60%), which is a common problem in long term retrospective studies. Over a wide range of prior-values, the Dirichlet process mixture model for selecting "survival experts" finds two large and highly stable clusters.

In order to externally validate these clusters, we analyze the survival of the underlying patient populations by way of classical Kaplan-Meier plots, see Figure 5.7. It can be easily inferred that the survival experiences of patients belonging to the two clusters differ significantly, with the "high-risk" cluster basically containing all patients who die early. In Figure 5.6 the interaction patterns within the two clusters are shown as lines connecting

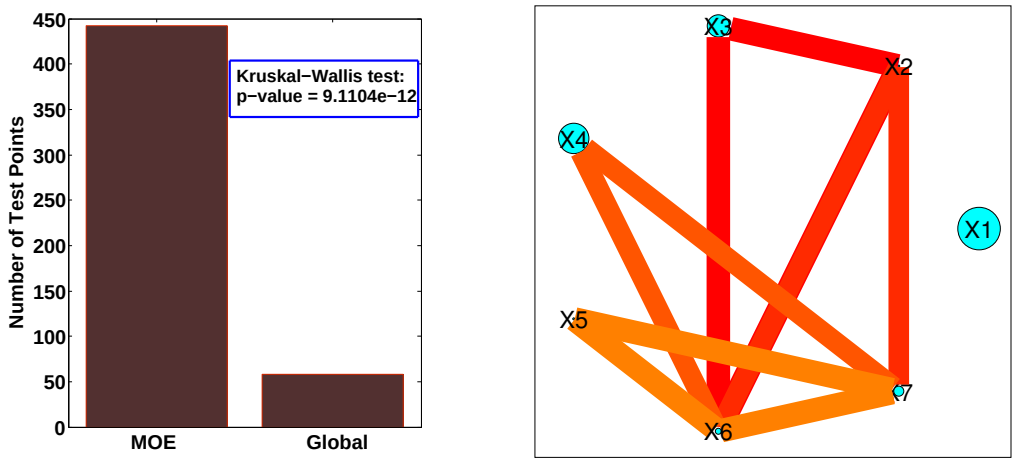


Figure 5.5.: **Left:** The actual number of points in the test set which scored better in a particular model (442 for a MOE model Vs 58 for a single cluster model) based on the likelihood scores. Results of the Kruskal-Wallis rank test also validated this observation with a p-value $\ll 0.001$. **Right:** Results of the significant interactions found for a single cluster model are shown. Some of the key factors are not identified along with existence of many false-positives.

pairs or triplets of markers, where the line width encodes the significance in terms of posterior quantiles which do not contain zero.

The high-risk patient cluster is characterized by a global under-expression of **ER** and over-expression of basically all other markers, in particular **KPNA2**, **CK5/6** and **HER2**. Over-expression of the latter two markers clearly identifies this cluster as a collection of *basal*- and **HER2**-type breast-cancer patients. The occurrence of **KPNA2** in the high-risk group is also in accordance with previous studies: **KPNA2** is a member of the karyopherin (importin) family, which is part of the nuclear transport protein complex. **KPNA2** over-expression has been shown in several gene expression signatures in breast cancer and other cancer types. **KPNA2** over-expression has been previously identified as a possible prognostic marker in breast cancer [39].

The Bayesian grouped variable selection framework for survival regression detects several strong higher-order interactions. Interpreting these interaction terms can be a complex problem, but a close analysis of the contrast codes and the sign of the regression coefficients shows that the weak prognosis of members in this cluster is dominated by some of the combinations, details in Table 5.1 where ↘ means under-expression and ↗ over-expression.

The observation that high-order interaction terms seem to be even more indicative than the individual main effects is a highly interesting result of this study which may lead to the definition of novel prognostic markers for better differentiation between high-risk patients. These new hypothetical compound-markers are being tested by our medical partners.

The low-risk cluster has a clear *luminal*-type signature (strong **ER** response). Hardly any significant patterns can be identified which, however, is quite understandable by noticing that the survival curve is almost flat for these patients: in the proportional hazards

Table 5.1.: The table shows the individual expression signs of all the significant interaction terms row-wise where ↘ means under-expression and ↗ over-expression. Each row denotes an interaction term and the number of elements in each row signify its order.

ER	↘				
ER	↘	CK5/6	↘		
KPNA2	↘	p53	↘	Collagen-VI	↘
ITIH5	↗	HER2	↗		
ER	↘	Collagen-VI	↘	HER2	↗
ER	↘	KPNA2	↘	ITIH5	↗
ER	↘	p53	↗	CK5/6	↘
ER	↘	KPNA2	↘	Collagen-VI	↘

model the individual variables influence the "passage of time", and a flat curve basically means that there is almost no intra-class variation that could be explained by individual variable effects.

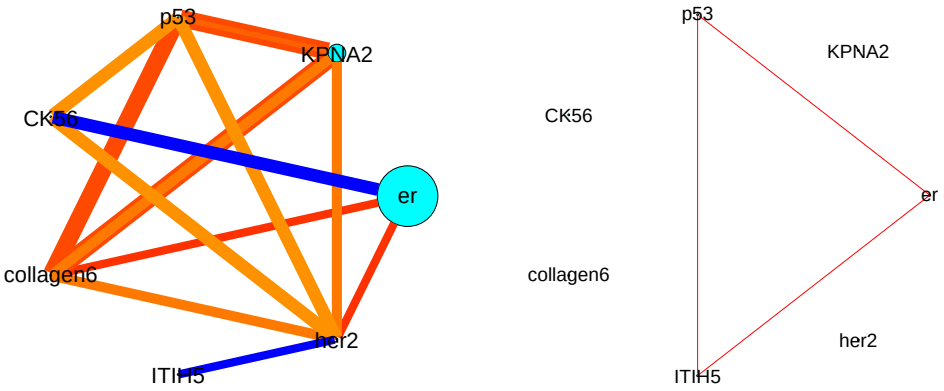


Figure 5.6.: Identified interaction patterns for the high-risk group (left) and the low-risk group (right). The size of the circles indicates the estimated significance of the main effects. For instance, the largest circle for ER means that the 0.9 posterior quantile does not contain zero. Correspondingly, the line-width of the interactions (blue lines: 1st-order, reddish triangles: 2nd-order) indicates their significance.

5.6 Summary

In this chapter, we have extended our Bayesian framework to the Weibull model which is extensively used for survival analysis. We have combined this extension with clustering by

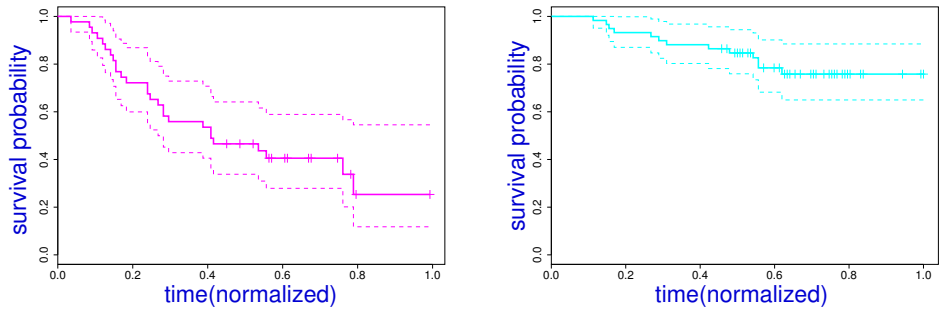


Figure 5.7.: Kaplan-Meier plots for the high-risk group (left) and the low-risk group (right). The high-risk group contains a large number of patients, who die early.

constructing a Bayesian survival infinite mixture-of-experts model which extends classical approaches by including feature selection for contrast-coded categorical variables. Random links and a mixture-of-experts architecture allow the underlying parametric survival model to be highly flexible. The inherent clustering property of the final model makes it possible to identify patient sub-groups which are homogeneous with respect to the effect of the predictor variables on the survival pattern. The use of our Bayesian grouped-variable selection framework allows the selection of factors and interactions specific to each cluster.

Applied to survival data from a breast cancer study, the model identified two stable patient clusters that show a clear distinction in terms of survival probability. Several strong high-order interactions between marker proteins were detected which may lead to the definition of novel prognostic markers.

So far we have used thresholding as an extra step to produce a sparse set of variables. Since producing a point estimate can be important from an application perspective, there is still a need for having a more principled way of obtaining a sparse point estimate which can then automatically lead to variable selection. In the next chapter, we will show how the existing algorithm can be extended via simulated annealing in order to produce such an estimate.

6

Point Estimate via Simulated Annealing

6.1 Variable Selection

In the context of variable selection in a linear regression problem, one of the frequent requirements is to produce a sparse point estimate with respect to the regression coefficients. From an application point of view, the domain practitioners often find such sparse point estimates useful in designing further experiments. So far we have developed a Bayesian approach for sparse regression for grouped variables with the capability to summarize the posterior distribution over the regression coefficients with estimates for quantities like the first and second moments. However, we still do not obtain a truly sparse estimate, since the expectation of the distribution over regression coefficients is not sparse.

From the perspective of the design of the overall Bayesian framework, our overall goal is also to create an omnibus framework which can estimate various quantities from the posterior distribution like expectation, variances and mode. An example of such an omnibus framework is defined in [58] where the focus is on logistic regression models. With this goal in mind, together with application requirements, we would additionally like to generate a sparse point estimate from our framework for variable selection.

So far, we have used heuristics like thresholding of the significance estimates of the regression coefficients for variable selection. Such thresholding techniques are commonly used by sparse Bayesian methods for generating a truly sparse point estimate (see [20]). In this chapter, we offer a more principled approach to additionally generate a truly sparse point estimate via **Simulated Annealing**(SA) with minimal changes to the already described framework for sparse regression. We will first describe the general concept of simulated annealing followed by its application to single variable selection in the context of the framework we have described so far. We will then generalize this idea to grouped-variable selection and to generalized linear models. This will be followed by experiments to illustrate the usefulness of this extension.

6.2 Simulated Annealing

Simulated annealing (as defined in [59] and [60]) is a stochastic search procedure for obtaining the global optimum of a given function usually described over a discrete domain. The concept is derived from annealing in metallurgy, which involves heating and controlled

cooling of a material in order to increase the size of its crystals and reduce their defects. The heating process causes atoms to change their initial configuration (which can be seen as some local minima of internal energy) and allows them to investigate new positions. The controlled slow cooling process which follows, then gives the atoms a chance to transition to states of lower internal energy than the initial one.

Translating this idea to find the optimal value of a probability distribution, samples are sequentially generated from the probability distribution over the variables, parameterized by a computational temperature parameter T, based on a cooling schedule for temperature. The cooling schedule is a function which allows the slow reduction of temperature over the iterations. The annealing process depends on the cooling schedule to be slow enough for it to reach the optimal value. The choice of the cooling schedule depends on the nature of the probability distribution under consideration.

For example, if our target distribution is $f(x)$, then the parameterized distribution is proportional to $f(x)^{\frac{1}{T}}$. We generate samples from this parameterized distribution, starting with the original distribution ($T = 1$), and slowly reduce T based on a cooling schedule. In each iteration, a sample point is generated based on the previous sample point using a transition probability distribution which is parameterized by T. As T is reduced, the new sample point is concentrated more around the neighborhood of the previous sample point. Asymptotically, under certain mild conditions, this results in the sampling process converging to the set of global maxima of the target distribution $f(x)$. A key point to note here is that unlike greedy methods for finding optimal values which allow only "downhill" updates, simulated annealing is stochastic in nature and hence allows "uphill" updates as well, giving the procedure a chance to escape local minima. This can be useful especially when dealing with multi-modal distributions. A generic Gibbs sampling algorithm for SA is as follows:

Algorithm 5 Example - Gibbs Sampling for Simulated Annealing

1: **Goal:** To find the mode of the distribution $P(x_1, x_2, x_3)$.
2: **Initialize:** Initialize values x_1^0, x_2^0, x_3^0.
3: Draw samples from the parameterized distribution $P(x_1, x_2, x_3)^{\frac{1}{T}}$ using Gibbs sampling
4: $T = 1$
5: **for** $m = 1$ to AnnealingIter **do**
6: Sample $x_1^m \sim P(x_1|x_2, x_3)^{\frac{1}{T}}$.
7: Sample $x_2^m \sim P(x_2|x_1, x_3)^{\frac{1}{T}}$.
8: Sample $x_3^m \sim P(x_3|x_1, x_2)^{\frac{1}{T}}$.
9: Decrease T according to a cooling schedule function $f_{cool}(\cdot)$, $T = f_{cool}(m)$
10: **end for**

Also Figure 6.1 shows the effect of temperature on a normal distribution. We observe that the decrease in temperature leads to the increase in concentration of mass at the mode. This translates to the effect of samples getting more concentrated near the mode as the chain progresses with decreasing temperature.

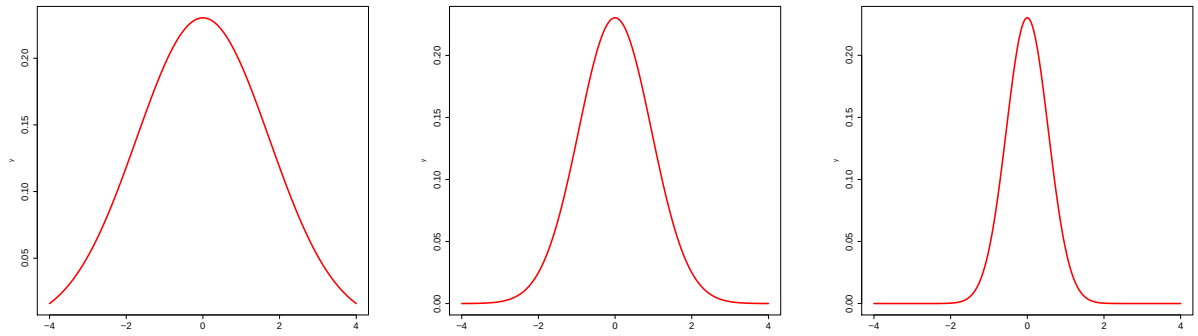


Figure 6.1.: The effect of temperature on a normal distribution. The mass gets more concentrated near the mode as the temperature is decreased from left to right.(Left: $T = 1$, Center: $T = 0.5$, Right: $T = 0.1$)

Convergence of Simulated Annealing. Although standard results exist for discrete domains for the asymptotic convergence of this procedure to the set of global maxima, it is much harder to prove convergence for continuous spaces in general. However, there is existing literature for convergence results for specific cases. For example, convergence of annealing in continuous cases for a specific version of the Metropolis sampler is shown by [61]. The work in [62] uses Foster-Lyapanov criteria for proving convergence for non-compact spaces under specified conditions. Other results involve proving convergence under a suitable hypothesis for diffusion processes (see [63];[64]) and analyzing their discretized versions [65]. Although not formally shown here, empirical observations indicate the convergence of Gibbs sampling in our experiments with grouped variable selection. For the rest of the chapter, we proceed with the assumption of convergence of the Gibbs sampler for annealing.

6.3 Extension to Bayesian Sparse Variable Selection

Based on the above description, we now propose the following extension to the Bayesian variable selection model in order to produce a sparse point estimate. We will first focus on the simpler case of single variable selection ($p_g = 1 \ \forall \ g$) in linear regression and then generalize the extension to grouped-variable selection. The posterior distribution that was described for the Bayesian variable selection problem consisted of variables $\boldsymbol{\beta}$, ρ, σ, α and Λ. As discussed earlier, the parameters ρ and σ play the role of the Lagrangian parameter $\left(= \sqrt{\rho\sigma^2}\right)$ in the classical Lasso and α controls the sparsity inducing properties of the prior over $\boldsymbol{\beta}$ ($\alpha = 1$ is the Lasso case).

To additionally generate a sparse $\boldsymbol{\beta}$ estimate, we extend this model to generate a MAP estimate of the joint posterior distribution over the parameters via simulated annealing. We then formally justify that this MAP estimate is sparse in $\boldsymbol{\beta}$. We have earlier shown σ^2

and ρ together play the role of the Lagrangian parameter in the classical Lasso and can be viewed as model selection parameters. Since they are model selection parameters, finding a MAP estimate over them would be sensitive to the specification of the prior distribution over the model selection parameters and can produce undesirable results. To avoid this undesired effect of the model selection parameters, we instead estimate the expected values of σ^2 and ρ from the samples generated by Algorithm 1 and fix it before proceeding to find the MAP estimate over the remaining variables $\boldsymbol{\beta}$ and Λ. This is justified as a model selection step after which one can find the optimal parameter values for the chosen model.

It is, however, important to note that finding a MAP solution over the joint posterior distribution over $\boldsymbol{\beta}$ and Λ is different from finding the MAP using the marginal prior of $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}|\alpha, \rho, \sigma^2)$(with Λ integrated out). We denote this marginal prior as P_M:

$$\begin{aligned} p(\boldsymbol{\beta}|\sigma^2, \alpha, \rho) &\propto \prod_{g=1}^{d} \int N(\beta_g|0, \lambda_g^2\sigma^2 I)\text{Gamma}(\lambda_g^2|\alpha, \frac{\rho}{2})d\lambda_g^2 \\ &= P_M. \end{aligned} \tag{6.1}$$

Using P_M we can obtain the MAP estimate for the "original" formulation of the problem i.e. the Bayesian Lasso:

$$p(\boldsymbol{\beta}|\boldsymbol{X}, \bullet) \propto \mathcal{L}(\boldsymbol{\beta})P_M, \tag{6.2}$$

where P_M has already shown to be sparsity inducing for $\alpha \leq 1$. Hence $\alpha = 1$ forms an upper bound below which the solutions of the MAP estimation problem in eq. (6.2) tends to be sparse. In the next section, we show that the same holds true for the joint posterior as well, with an adjusted value of α.

Following the simulated annealing procedure for our model, we introduce a computational temperature parameter T for the posterior distribution over $(\boldsymbol{\beta}, \Lambda)$ with all the other variables fixed:

$$\begin{aligned} p(\boldsymbol{\beta}, \Lambda|\boldsymbol{X}, \boldsymbol{y}, \sigma^2, \alpha, \rho, T) = \quad & Z(T) \overbrace{\left(\prod_{i=1}^{n} N(y_i|\boldsymbol{x}_i^t\boldsymbol{\beta}, \sigma^2)\right)^{\frac{1}{T}}}^{Likelihood} \\ & \cdot \overbrace{\left(\prod_{g=1}^{d} N(\boldsymbol{\beta}_g|0, \lambda_g^2\sigma^2 I)\right)^{\frac{1}{T}} \left(\prod_{g=1}^{d} \text{Gamma}(\lambda_g^2|\alpha, \frac{\rho}{2})\right)^{\frac{1}{T}}}^{JointPrior}, \end{aligned} \tag{6.3}$$

where $Z(T)$ is a function in T introduced for normalization purposes.

Since all the posterior conditional distributions retain the same standard forms as in the original problem with only a change in the parameters, we again use a Gibbs sampling strategy to sample from this parameterized distribution. The conditional distributions are as follows:

$$p(\lambda_g^2|\bullet) \sim \text{GIG}\left(\frac{\alpha - 1.5 + T}{T}, \frac{a_g}{T}, \frac{b_g}{T}\right), \tag{6.4}$$

$$p(\boldsymbol{\beta}|\bullet) \sim N(\boldsymbol{\beta}|\tilde{\boldsymbol{\mu}}, \sigma^2 T\widetilde{\Sigma}), \tag{6.5}$$

where $a_g = \rho$, $b_g = \frac{\|\beta_g\|_2^2}{\sigma^2}$, $\widetilde{\Sigma} = (X^tX + \Lambda^{-1})^{-1}$, $\tilde{\boldsymbol{\mu}} = \widetilde{\Sigma}X^t\boldsymbol{y}$ and Λ is a $(d \times d)$ diagonal matrix consisting of λ_g^2's as diagonal elements, where $g \in \{1, .., d\}$ and $\bullet$ denotes the rest of the parameters. The detailed sampling steps for the modified Gibbs sampling algorithm are described in Algorithm 6.

Algorithm 6 Gibbs Sampling for Simulated Annealing

1: **Input:** n observations $D = (\boldsymbol{x}_i, y_i)_{i=1}^n$.
2: **Initialize:** Parameters $\boldsymbol{\beta}, \sigma^2, \rho, \Lambda, \alpha$.
3: Draw samples from the posterior of joint distribution $p(\boldsymbol{\beta}, \sigma^2, \rho, \Lambda, |\alpha, D, T)$ by drawing from the conditionals.
4: T = 1
5: **for** $m = 1$ to BayesIter **do**
6: Sample $\rho|\boldsymbol{\beta}, \sigma^2, \Lambda, \boldsymbol{y}, \alpha, D$ - from a gamma distribution.
7: Sample $\Lambda\ |\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{y}, \alpha, D$ - from a generalized inverse Gaussian distribution.
8: Sample $\boldsymbol{\beta}, \sigma^2\ |\rho, \Lambda, \boldsymbol{y}, \alpha, D$ - σ^2 is sampled from an inverse-chi square distribution and $\boldsymbol{\beta}$ conditioned on σ^2 from a multivariate normal distribution.
9: **end for**
10: Fix σ^2 and ρ to the expected values based on posterior samples.
11: **for** $m = 1$ to AnnealingIter **do**
12: $T = f_{cool}(m, T)$ - Cooling schedule for T based on the current iteration number.
13: Sample $\Lambda\ |\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{y}, \alpha, D, T$ - from a generalized inverse Gaussian distribution.
14: Sample $\boldsymbol{\beta}\ |\rho, \sigma^2, \Lambda, \boldsymbol{y}, \alpha, D, T$ - $\boldsymbol{\beta}$ from a multivariate normal distribution.
15: **end for**

So far what we have obtained through this algorithm is the MAP estimate for the joint posterior distribution over $(\boldsymbol{\beta}, \Lambda)$. It is non-trivial to assume that this would also result in an estimate which is sparse in $\boldsymbol{\beta}$. Next, we discuss the properties of this MAP estimate with respect to sparsity of $\boldsymbol{\beta}$ and also its connection to the MAP estimate of the original problem. i.e. MAP of $\boldsymbol{\beta}$ with Λ integrated out. This is done using the concept of variational formulation which is briefly described in the next section.

6.4 Sparsity Properties of the Joint MAP Estimate

In this section we will analyze the sparsity properties of the joint MAP estimate of $(\boldsymbol{\beta}, \Lambda)$. We will first briefly explain the idea of variational methods which will be used later to show that the joint MAP estimate is sparse under specific conditions. We will also assume that the likelihood function is a normal distribution.

Variational Methods. The idea of variational methods has its roots in calculus of variations [66]. In various applications, these methods are used to rewrite a complex function

into a simpler function by introducing additional auxiliary variables. We describe the basic idea with an example, in the context of the usage in the next section.

We consider the example of a logarithm function as described in [67] and express it in a variational form using a parameter λ:

$$\ln(x) = \min_{\lambda}\{\lambda x - \ln\lambda - 1\}. \tag{6.6}$$

Hence the logarithm function has been re-expressed using a minimization of a function of λ for each fixed value of x. At each value of λ, we obtain a line over x and over all λs we obtain a whole set of lines which form an upper envelope of logarithm function, which means that for any x, all the lines of the envelope lie above x or:

$$\ln(x) \leq \lambda x - \ln\lambda - 1. \tag{6.7}$$

Hence, through the λs we obtain an upper bound of the logarithm at x. Also, as a result, a non-linear function has been converted to a set of linear functions. In the next section, we shall use the notion of variational methods to show the equivalence of two functions, where one is a variational form of the other. This would further imply that the optimization of either functions with respect to the given parameters results in the same solution.

Joint Vs Single MAP Estimation Problems. We begin by defining the two MAP estimation problems related to this Bayesian variable selection model. The first one is the joint MAP estimate of the regression coefficients and the auxiliary variables:

Definition 1. *MAP_1:* $\arg\max_{\boldsymbol{\beta},\Lambda} p(\boldsymbol{\beta}, \Lambda|\bullet)$.

The annealing algorithm defined in the previous section is based on this joint MAP estimate. The second estimation problem is based on obtaining a MAP estimate from the formulation of the single variable selection problem defined in eq. (2.36) and eq. (2.37) after integrating out the auxiliary variables Λ:

Definition 2. *MAP_2:* $\arg\max_{\boldsymbol{\beta}} \int p(\boldsymbol{\beta}|\Lambda, \bullet)p(\Lambda|\bullet)d\Lambda$.

For $\alpha = 1$, MAP_2 is a representation of the Lasso and hence it will tend to produce estimates which are sparse in $\boldsymbol{\beta}$. Also, this defines a threshold on the value of $\hat{\alpha}$ $(= 1)$ below which the solutions of MAP_2 will tend to be sparse. Our goal is to show that MAP_1 for $\alpha = 1.5$ is equivalent to MAP_2 for $\alpha = 1.0$ and hence will also tend to produce sparse solutions. We will also show that the value of $\alpha = 1.5$ is the adjusted threshold value for MAP_2 below which the solutions will tend to be sparse.

Sparsity of MAP_2 and Equivalence to MAP_1

Proposition 1. *The solution of MAP_1 for $\alpha = 1.5$, and the solution of MAP_2 for $\alpha = 1$ coincide with respect to $\boldsymbol{\beta}$.*

Proof. The proof is based on showing that MAP_1 at $\alpha = 1.5$ is a variational formulation of MAP_2 at $\alpha = 1$. Consider the joint posterior of $\boldsymbol{\beta}$ and Λ for a generic α value:

$$\begin{aligned} p(\boldsymbol{\beta}, \Lambda|\bullet) \propto & \mathcal{L}(\boldsymbol{\beta}) N(\boldsymbol{\beta}|0, \sigma^2\Lambda) \\ & \cdot \prod_{g=1}^{d} \text{Gamma}(\lambda_g^2|\alpha, \frac{\rho}{2}), \end{aligned} \tag{6.8}$$

where$\mathcal{L}()$ is the likelihood function. For $\alpha = 1.5$:

$$p(\boldsymbol{\beta}, \Lambda|\bullet) \propto \mathcal{L}(\boldsymbol{\beta}) \prod_{g=1}^{d} \exp[-0.5(\frac{\beta_g^2}{\sigma^2\lambda_g^2} + \lambda_g^2\rho)]. \tag{6.9}$$

Taking negative log likelihood:

$$\mathcal{C}(\boldsymbol{\beta}, \Lambda) = 0.5 \sum_{g=1}^{d} (\frac{\beta_g^2}{\sigma^2\lambda_g^2} + \lambda_g^2\rho) - \ln \mathcal{L}(\boldsymbol{\beta}), \tag{6.10}$$

where $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ is the resulting cost function which needs to be minimized (equivalent to $p(\boldsymbol{\beta}, \Lambda)$ being maximized) ignoring the constant terms. First consider fixing $\boldsymbol{\beta}$ and finding the optimal Λ for a fixed $\boldsymbol{\beta}$ with the following motivation. We consider the joint cost function over $(\boldsymbol{\beta}, \Lambda)$ to be a variational form of some function over $\boldsymbol{\beta}$ and the goal is to find this "original" function. To find the optimal value of Λ, we minimize $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ for a fixed $\boldsymbol{\beta}$ by taking partial-derivatives with respect to λ_g's separately since all the λ_g can be optimized separately:

$$\begin{aligned} & \frac{\partial \mathcal{C}(\boldsymbol{\beta}, \Lambda)}{\partial \lambda_g} = -\frac{\beta_g^2}{\sigma^2\lambda_g^3} + \lambda_g\rho = 0 \\ & \implies \hat{\lambda_g^2} = \sqrt{\frac{\beta_g^2}{\sigma^2\rho}}, \end{aligned} \tag{6.11}$$

where $\hat{\lambda_g^2}$ denotes the optimal value of λ_g^2. Hence this implies that the function $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ is an envelope function for $\mathcal{C}(\boldsymbol{\beta}, \hat{\Lambda})$ at every $\boldsymbol{\beta}$ where $\hat{\Lambda} = \{\hat{\lambda_g^2}\}_{g=1}^{d}$.

Using this optimal value, we replace $\hat{\Lambda}$ in $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ to get a function only in $\boldsymbol{\beta}$:

$$\mathcal{C}(\boldsymbol{\beta}) = -\ln \mathcal{L}(\boldsymbol{\beta}) + \sqrt{\frac{\rho}{\sigma^2}} \sum_{g=1}^{d} |\beta_g|. \tag{6.12}$$

Based on the above equation, minimizing $C(\boldsymbol{\beta})$ translates exactly to the Lasso optimization problem (or the log counterpart of the Bayesian variable selection problem for $\alpha = 1$) or equivalently MAP_2. Solving for the optimal value for this function gives us a sparse MAP estimate of $\boldsymbol{\beta}$. Hence, we have shown that, at $\alpha = 1.5$, MAP_1 is a variational formulation

of MAP_2 at $\alpha = 1$ and solving for $\boldsymbol{\beta}$ in either case tends to produce a sparse point estimate. □

We also observe such a similar variational formulation of the Group-Lasso has been used in the context of multiple kernel learning in [23]. To summarize, we have shown that MAP_2 for $\alpha = 1$ is equivalent to MAP_1 for $\alpha = 1.5$. We now show that this value of 1.5 can be viewed as an upper bound on α, below which the solutions for MAP_1 tend to be sparse in $\boldsymbol{\beta}$.

Upper Bound for α. The arguments above that were used for the specific case of $\alpha = 1.5$ can be generalized for all values of α. As before, we derive the expression for $C(\boldsymbol{\beta}, \Lambda)$ for a general α from $p(\boldsymbol{\beta}, \Lambda|\bullet)$:

$$\begin{aligned} p(\boldsymbol{\beta}, \Lambda|\bullet) \propto & \mathcal{L}(\boldsymbol{\beta}) N(\boldsymbol{\beta}|0, \sigma^2\Lambda) \cdot \prod_{g=1}^{d} \text{Gamma}(\lambda_g^2|\alpha, \frac{\rho}{2}) \\ \propto & \mathcal{L}(\boldsymbol{\beta}) \prod_{g=1}^{d} \left(\exp[-0.5(\frac{\beta_g^2}{\sigma^2\lambda_g^2} + \lambda_g^2\rho)](\lambda_g^2)^{\alpha-1.5} \right). \end{aligned} \tag{6.13}$$

Taking negative log likelihood we get:

$$\mathcal{C}(\boldsymbol{\beta}, \Lambda) = 0.5 \sum_{g=1}^{d} \left(\frac{\beta_g^2}{\sigma^2\lambda_g^2} + \lambda_g^2\rho - 2(\alpha - 1.5)\ln(\lambda_g^2) \right) - \ln \mathcal{L}(\boldsymbol{\beta}). \tag{6.14}$$

As before we find the optimal value of each λ_g for a fixed $\boldsymbol{\beta}$. This gives us:

$$\begin{aligned} \frac{\partial \mathcal{C}(\boldsymbol{\beta}, \Lambda)}{\partial \lambda_g} &= -\frac{\beta_g^2}{\sigma^2\lambda_g^3} + \lambda_g\rho - \frac{2(\alpha - 1.5)}{\lambda_g} = 0 \\ \implies \hat{\lambda}_g^2 &= \frac{(\alpha - 1.5) + \sqrt{(\alpha - 1.5)^2 + b_g\rho}}{\rho}, \end{aligned} \tag{6.15}$$

where b_g is defined as before. Note that for $\alpha = 1.5$, we get back eq. (6.11). Similar to eq. 6.12, we derive $\mathcal{C}(\boldsymbol{\beta})$ by replacing λ_g^2 with its optimal value $\hat{\lambda}_g^2$, which results in a more complicated expression:

$$\mathcal{C}(\boldsymbol{\beta}, \Lambda) = 0.5 \sum_{g=1}^{d} \left(\frac{\beta_g^2}{\sigma^2\hat{\lambda}_g^2} + \hat{\lambda}_g^2\rho - 2(\alpha - 1.5)\ln(\hat{\lambda}_g^2) \right) - \ln \mathcal{L}(\boldsymbol{\beta}). \tag{6.16}$$

To analyze the properties of this function, we reformulate $\mathcal{C}(\boldsymbol{\beta})$ in probabilistic terms by reversing the negative logarithm operation that was done earlier to obtain this function from a probability distribution. By exponentiating $(-\mathcal{C}(\boldsymbol{\beta}))$, we can express the function in terms of a posterior distribution of $\boldsymbol{\beta}$ broken up into the product the likelihood and

prior terms:

$$p(\boldsymbol{\beta}|\hat{\Lambda}, \bullet) \propto \mathcal{L}(\boldsymbol{\beta}) \prod_{g=1}^{d} \left(\exp \left[-0.5 \left(\frac{\beta_g^2}{\sigma^2 \hat{\lambda}_g^2} + \hat{\lambda}_g^2 \rho \right) \right] (\hat{\lambda}_g^2)^{\alpha - 1.5} \right) \tag{6.17}$$
$$\propto \mathcal{L}(\boldsymbol{\beta}) P_C,$$

where $P_C \propto p(\boldsymbol{\beta}|\hat{\Lambda}, \alpha, \sigma^2, \rho)$ which can be interpreted as a prior for $\boldsymbol{\beta}$ conditioned on the optimal value of the auxiliary variables Λ, which is clearly different from P_M where Λ was integrated out.

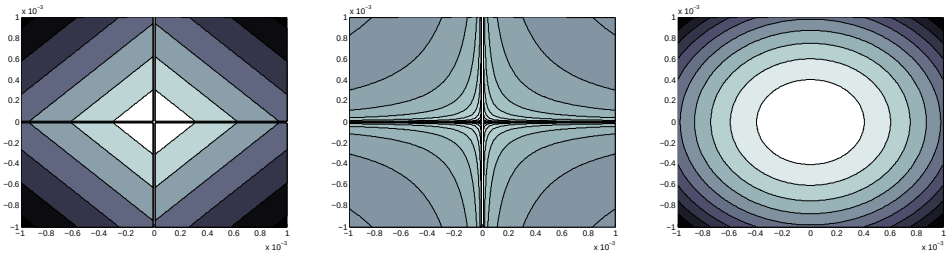


Figure 6.2.: **Left:** Plot of P_C for $\alpha = 1.5$ which resembles the Lasso constraint. **Center:** Plot of P_C for $\alpha = 1.0$ which is also observed to be a sparsity inducing prior. **Right:** Plot of P_C for $\alpha = 2.0$ which resembles a normal distribution.

To understand the nature of the posterior over $\boldsymbol{\beta}$, we plotted the prior P_C in two dimensions and for three different α values (1, 1.5 and 2), giving us the resulting plots in Figure 6.2. At value 1.5, P_C takes the form of a product of Laplace distributions (left panel of Figure 6.2), which is a sparsity inducing prior. For $\alpha = 1$, P_C is still a sparsity inducing prior (center panel of Figure 6.2) and this is observed in general for values ≤ 1.5. Further, for $\alpha > 1.5$, P_C becomes a non-sparsity inducing prior and resembles the Gaussian distribution (see right panel of Figure 6.2).

Based on these plots, we observe that $\alpha = 1.5$ is a threshold value below which MAP_1 tends to be sparse in $\boldsymbol{\beta}$. For the original problem of Bayesian variable selection i.e. MAP_2, the threshold for sparsity inducing priors is 1.0. Hence to apply simulated annealing for finding the solution to MAP_2, we simply have to adjust the threshold to 1.5 for inducing sparsity in the solution. In summary, we have shown that the joint MAP estimate produced by simulated annealing tends to be sparse in the regression coefficients for $\alpha \leq 1.5$ and hence can be used, in addition to the first and second moments, to summarize the posterior over the regression coefficients. As a result, it offers a more principled approach for obtaining truly sparse estimates for variable selection in the Bayesian regime as opposed to heuristics like thresholding.

6.5 Further Extensions

Although the above discussions were centered around single variable selection applied to a linear regression problem, these arguments can be extended to other cases. There are two aspects of this extension. The first aspect deals with extending the single variable selection case to grouped variable selection, where sparsity is desired over predetermined groups of regression coefficients as opposed to individual coefficients. Similar to the single variable case, the goal is to generate a sparse point estimate for the Bayesian grouped variable selection framework that we have defined so far. The second aspect deals with extending the model from the perspective of the likelihood term which was till now assumed to be a normal distribution. The goal is to generate a MAP estimate when the grouped variable selection framework is extended for generalized linear models resulting in a changed likelihood term.

Grouped Variable Selection. For extending all the previous arguments regarding MAP estimates to grouped variable selection, we use the prior defined in the previous chapter:

$$\begin{aligned} p(\boldsymbol{\beta}, \Lambda|\sigma^2, \alpha, \rho) \propto & \prod_{g=1}^{G} N(\boldsymbol{\beta}_g|0, \lambda_g^2\sigma^2 I) \\ & \cdot \prod_{g=1}^{G} \text{Gamma}(\lambda_g^2|\frac{p_g+1}{2}\alpha, \frac{p_g\rho}{2}), \end{aligned} \tag{6.18}$$

where as before $\boldsymbol{\beta}$ is divided into G groups/sub-vectors $\boldsymbol{\beta}_g$, where each group g is of size p_g. As we saw in the last chapter, similar to the single variable selection case, all the posterior conditional distributions are of standard form and hence Gibbs sampling is applied using Algorithm 1. After the model selection step (fixing ρ and σ), the annealing is carried out in the same way as in Algorithm 6, since the conditionals are again of the same form. The conditional distributions for the grouped-variable selection case are almost identical:

$$p(\lambda_g^2|\bullet) \sim \text{GIG}\left(\frac{\alpha - 1.5 + T}{T}, \frac{a_g}{T}, \frac{b_g}{T}\right), \tag{6.19}$$

$$p(\boldsymbol{\beta}|\bullet) \sim N(\boldsymbol{\beta}|\tilde{\boldsymbol{\mu}}, \sigma^2 T\widetilde{\Sigma}), \tag{6.20}$$

where $a_g = \rho$, $b_g = \frac{\|\boldsymbol{\beta}_g\|_2^2}{\sigma^2}$, $\widetilde{\Sigma} = (X^tX + \Lambda^{-1})^{-1}$, $\tilde{\boldsymbol{\mu}} = \widetilde{\Sigma}X^t\boldsymbol{y}$ and the only change from the single variable case is the Λ matrix which is defined as Figure 3.2.

The justification of the sparsity of $\boldsymbol{\beta}$ can also be shown in the same way as in the single variable selection case. Using eq. (6.18) as prior, the posterior can be expanded as follows:

$$p(\boldsymbol{\beta}, \Lambda|\bullet) \propto \mathcal{L}(\boldsymbol{\beta}) \prod_{g=1}^{G} \left((\lambda_g^2)^{(\frac{p_g+1}{2}\alpha - \frac{p_g}{2} - 1)} \exp\left[-0.5 \left(\frac{b_g}{\lambda_g^2} + \lambda_g^2 a_g \right) \right] \right), \tag{6.21}$$

where $a_g = p_g\rho$ and $b_g = \frac{||\boldsymbol{\beta}_g||^2}{\sigma^2}$.

Taking negative log likelihood:

$$\mathcal{C}(\boldsymbol{\beta}, \Lambda) = 0.5 \sum_{g=1}^{G} \left(\frac{b_g}{\lambda_g^2} + \lambda_g^2 a_g \right) - p_g^{'} \log \lambda_g^2) - \ln \mathcal{L}(\boldsymbol{\beta}), \tag{6.22}$$

where $p_g^{''} = \frac{p_g+1}{2}\alpha - \frac{p_g}{2} - 1$ and $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ is the resulting cost function which needs to be minimized (equivalent to $p(\boldsymbol{\beta}, \Lambda)$ being maximized) ignoring the constant terms. Following the proof in the previous section, we fix $\boldsymbol{\beta}$ and find the optimal Λ. Minimizing $\mathcal{C}(\boldsymbol{\beta}, \Lambda)$ for a fixed $\boldsymbol{\beta}$, we obtain the optimal value of each λ_g $\forall g$ as:

$$\hat{\lambda_g^2} = \frac{p_g^{''} + \sqrt{(p_g^{''})^2 + b_g a_g}}{a_g}. \tag{6.23}$$

Following the same argument as in the previous section, we reformulate $\mathcal{C}(\boldsymbol{\beta})$ by fixing Λ to $\hat{\Lambda}$. Then, to interpret the nature of the solution produced by this optimization problem, we again reformulate it in probabilistic terms as a product of likelihood and prior (P_C) as was done for the single variable case.

On similar lines, we obtain the threshold value for α below which the prior (P_C) is sparsity inducing. To find the threshold value of α in the grouped-variable case, we try to find the particular value for which the solution matches that of the Bayesian Group-Lasso. Hence we try to find that value for which the joint MAP estimation problem for $(\boldsymbol{\beta}, \Lambda)$ becomes a variational formulation of the Bayesian Group-Lasso. We obtain this threshold by setting $p_g^{''} = 0 \implies \alpha = \frac{p_g+2}{p_g+1}$. Since we have to consider a common α value for all the groups, setting $\hat{\alpha} = \min_g \left(\frac{p_g+2}{p_g+1} \right)$ establishes the threshold for the grouped-variable selection case below which the solution of the grouped variable version of MAP_1 will tend to be sparse in groups of regression coefficients. Since $\hat{\alpha} > 1$, setting $\alpha \leq 1$ ensures that the optimal value of $\boldsymbol{\beta}$ will tend to be sparse.

Generalized Linear Models. Till now, we have discussed annealing in the context of a Gaussian likelihood. A further extension to the annealing component is to apply it to other generalized linear models like the binomial model or the Poisson model. As described in Chapter 4, the Bayesian grouped-variable framework applied to generalized linear models involves the introduction of the auxiliary variable $\boldsymbol{\eta}$. As we have seen earlier, the introduction of $\boldsymbol{\eta}$ modifies the model such that the posterior over the variables is written as:

$$\begin{aligned} p(\boldsymbol{\beta}, \Lambda, \boldsymbol{\eta} | \sigma^2, \alpha, \rho) \propto \mathcal{L}(g^{-1}(\boldsymbol{\eta})) N(\boldsymbol{\eta} | X\boldsymbol{\beta}, \sigma^2 I) \prod_{g=1}^{G} N(\boldsymbol{\beta}_g | 0, \lambda_g^2 \sigma^2 I) \\ \cdot \prod_{g=1}^{G} \mathrm{Gamma}(\lambda_g^2 | \frac{p_g + 1}{2}\alpha, \frac{p_g \rho}{2}), \end{aligned} \tag{6.24}$$

where $g(\cdot)$ denotes the link function as defined previously and the likelihood functions that we consider are from the exponential family like the binomial and Poisson models that we considered in Chapter 4.

To apply Gibbs sampling to the un-annealed GLM version of the model, as described in Algorithm 2, the only change required in the algorithm is the sampling of $\boldsymbol{\eta}$ conditioned on other variables. The sampling of other variables does not change. For example, we have seen that the conditional posterior of $\boldsymbol{\eta}$ for the binomial probit model is a truncated normal distribution. Although the conditional posterior of $\boldsymbol{\eta}$ does not always take a standard form, suitable approximations can be applied based on the model under consideration as was observed in the Poisson regression case.

To apply the annealing step, there are two options while dealing with the variable $\boldsymbol{\eta}$. The first option involves jointly annealing $(\boldsymbol{\beta}, \Lambda, \boldsymbol{\eta})$ and hence finding a joint MAP estimate including $\boldsymbol{\eta}$. But since $\boldsymbol{\eta}$ is a stochastic link, this leads to an undesirable mixing problem in the sampler, and we observe that it shrinks $\boldsymbol{\eta}$ and hence $\boldsymbol{\beta}$ to zero. We choose a second option in which we use the same strategy as for the variables (σ^2, ρ). We fix $\boldsymbol{\eta}$ to its estimated expected value from the samples accumulated before the annealing step. Hence the annealing proceeds as before with the remaining two variables $\boldsymbol{\beta}$ and Λ and all our previous results regarding sparse estimates are valid.

6.6 Experiments

6.6.1. Lasso - Regression - Diabetes Dataset

To demonstrate the usefulness of adding a new estimator to the existing Bayesian framework for sparse regression, we use the diabetes dataset first used by [7] and later on used by other Lasso based algorithms for comparison. This data set is available as a part of the lars R package and also at (http://www-stat.stanford.edu/∼ hastie/Papers/LARS/). The data consists of $n = 442$ diabetes patients and $d = 10$ variables measured for each patient. The response is a measure of disease progression. The inference was first carried out using the optimization based algorithm in the LARS package. The solution paths of all the coefficients produced by the LARS package are shown in Figure 6.3 left panel. But as mentioned before, the optimization based framework does not provide any further information (like the variance estimates of the zero coefficients) regarding the posterior over the regression coefficients as opposed to the Bayesian framework. To obtain more posterior information for the same dataset, we ran the Gibbs sampling steps without annealing (i.e. till step 9 in Algorithm 6). The sampling was carried out for 5000 iterations. The cooling function is implemented by decreasing T every k iterations geometrically using the function $T^{'} = cT$, where $c < 1$. We tuned the cooling parameters based on the trace plot of the samples generated and fixed the value to $k = 50$ and $c = 1.1$. We then summarize the posterior with a box plot as shown in Figure 6.3 right panel. The box plot provides more detailed information about the variances of the regression coefficients. Also as mentioned earlier, the estimated mean is not sparse in $\boldsymbol{\beta}$. Also in Figure 6.4 left panel, we plot the significance levels of the coefficient values based on the significance

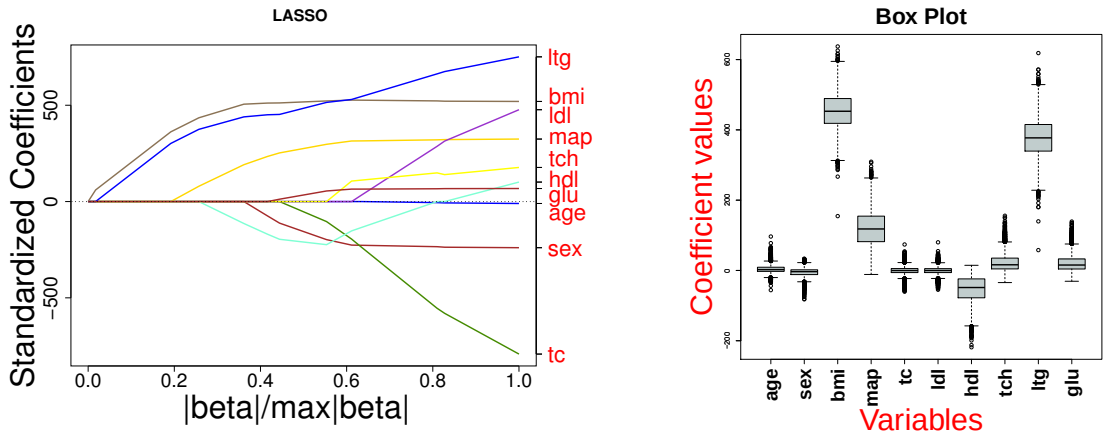


Figure 6.3.: **Left:** Plot of the Lasso solution path generated from the LARS R package which contains a standard Lasso implementation.**Right:** A box plot of the coefficient values calculated from the samples generated from the un-annealed part of Algorithm 6.

plot described in earlier chapters. Variable selection can be performed by thresholding at a particular level. The plot shows three such possible thresholds, which can result in different number of variables selected and it is unclear which threshold is better. To have a more principled approach to variable selection, we execute our proposed extension of the algorithm (simulated annealing) for about $10,000$ iterations to produce a truly sparse estimate of the regression coefficients which automates the variable selection step. The final sparse output is shown in the right panel of Figure 6.4.

Based on the above plots, we observe the advantages of the Bayesian framework which clearly provides more information regarding the posterior distribution of the regression coefficients. Additionally, our proposed extension provides a principled approach to variable selection in the Bayesian regime as opposed to heuristics like thresholding.

6.6.2. Flexible Sparsity Parameter - Toy Experiment

We designed a toy experiment for sparse regression in order to highlight the significance of having a flexible sparsity inducing parameter (α) for cases when the Lasso tends to select too many features. Selecting lesser features leads to a global shrinkage of all the regression coefficients which can in some case lead to a decrease in predictive power of the learnt model. The dataset that we generated consisted of $n = 100$ observations with $d = 50$. The regression coefficients that were set to non-zero (value $= 0.6$) were $\beta_4, \beta_{14}, \beta_{30}, \beta_{37}, \beta_{45}$. Based on the generated data, the annealing was first done using the Lasso threshold value ($\alpha = 1.5$), which resulted in the Lasso based MAP solution. The results shown in Figure 6.5 left panel clearly shows some extra false positives selections in terms of variable selection. This can easily be fixed by tuning of the ρ parameter for obtaining a sparser solution

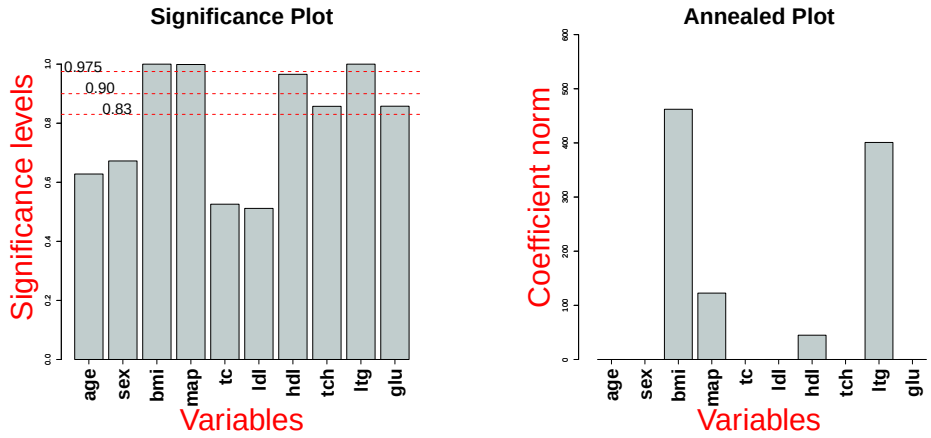


Figure 6.4.: **Left:** A significance plot for the regression coefficients indicating significance of regression coefficients as described in earlier chapters. **Right:** The plot of the norms of the coefficient values after annealing resulting a truly sparse estimate.

but this results in excessive global shrinkage of the true positives. As opposed to tuning the constraint parameter ρ, we ran the annealing for $\alpha = 1.0$ and obtained the true sparse pattern as shown in Figure 6.5 right panel without compromising excessively on the coefficient values. Additionally to compare predictive performance, we obtained the prediction error on a 1000 separate test sets of 1000 observations each. The prediction error was calculated based on the MAP estimate that was obtained in both cases. We observe that in 65.8% of the datasets, the case of $\alpha = 1.0$ performed better than the Lasso MAP estimate based on $\alpha = 1.5$.

6.6.3. Group Lasso - Classification - MEMset Donor Dataset

The analysis of DNA sequences to locate genes is an important task in genomics. Genes, however, do not necessarily occur as a continuous sequence in the DNA, but are separated by non-coding regions known as introns. Splice sites are regions in the DNA which separate the coding regions (exons) from introns. In particular, we analyze the donor splice site, which is marked by the $5'$ end (starting point) of an intron. For our analysis, we use the MEMset Donor dataset (freely available at http://genes.mit.edu/burgelab/maxent/ssdata/), which consists of 8415 true and 179438 false human donor sites. For our experiments, the data was balanced (see [22]) in both datasets to have an equal number of true and false splice site observations. Each instance of data consists of a sequence of DNA within a window of the splice site which consist of the last 3 positions of the exon (-3,-2,-1) and first 4 positions (2,3,4,5) of the intron (string of length 7). Hence these strings of length 7 are made up of 4 characters A, C, T, G, see [44] for details. Apart from the main effects (individual variables), the data is extended further to include 1st order (pairwise) interactions. Since the data is categorical, each interaction term is then coded with dummy

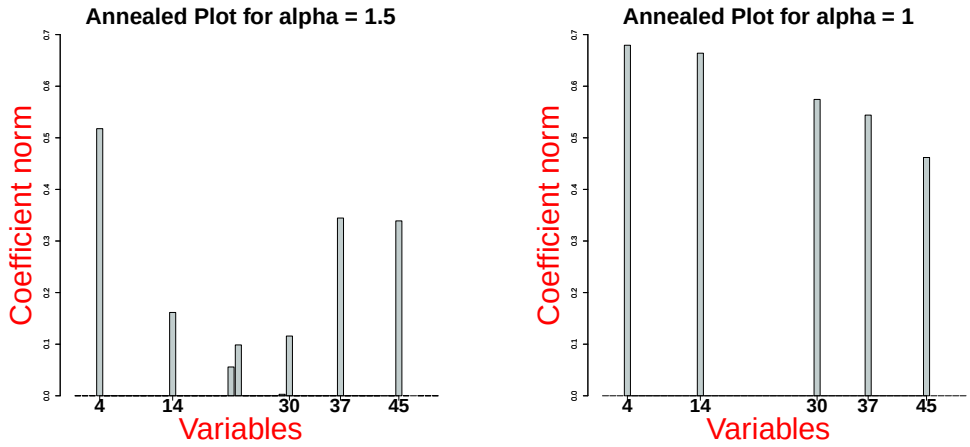


Figure 6.5.: **Left:** The results of running the Lasso ($\alpha = 1.5$) which produces a less sparse solution and also results in excessive shrinkage of coefficients. **Right:** The plot of the norms of the coefficient values after annealing was done with $\alpha = 1.0$ which resulted in detecting the true underlying sparse pattern without excessive shrinkage of the regression coefficients.

variables using a polynomial contrast code giving rise to a groups of variables with a design matrix of size 16830×211. The goal of the analysis is to identify the key interaction patterns with respect to the classification of true vs false splice sites for which we use a binomial probit model.

The first step involved executing the un-annealed version of our algorithm, by setting temperature $T = 1$, in order to obtain the estimates for the parameters ρ, σ^2 and $\boldsymbol{\eta}$. This involves executing Algorithm 6 till step 10. Figure 6.6 displays box plot of the group norms of the interactions for this part of the experiment.

After fixing ρ and $\boldsymbol{\eta}$, we apply simulated annealing in order to obtain estimates which are sparse in the regression coefficients. The results are described in Figure 6.6 which shows the interactions which were selected as a result of simulated annealing in two forms, a bar graph and an equivalent graph representation. Apart from some of the neighboring interactions within the intron and exon regions, we also observe a an *inter-region* interaction between (-1:2), which are the last position of the exon and first position of the intron respectively. This particular observation further validates the assertion made in [12], which does not find the inter-region interaction as important, but shows that inter-region interactions may have a role in solutions with the same (or ϵ-close) likelihood.

A standard optimization based Group-Lasso experiment was also executed on this dataset. The results in Figure 6.8 show the plot of the solution path i.e. the traces of the group norms under relaxation of the constraint κ. As before, the results fail to provide more information regarding the posterior distribution of the regression coefficients. The Bayesian framework, on the other hand, is able to provide posterior estimates of the first and second moments and now additionally a sparse point estimate based on the simulated

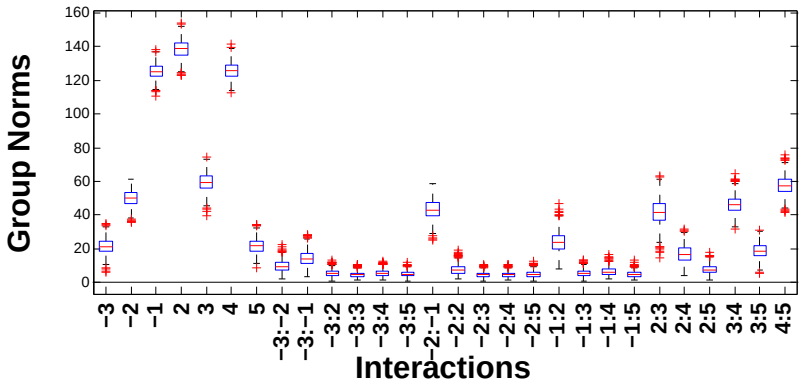


Figure 6.6.: Box plot of the group-norms of the regression coefficients using the samples generated by Gibbs sampling. The interactions include the individual terms denoting the window positions and pairwise interactions between these positions.

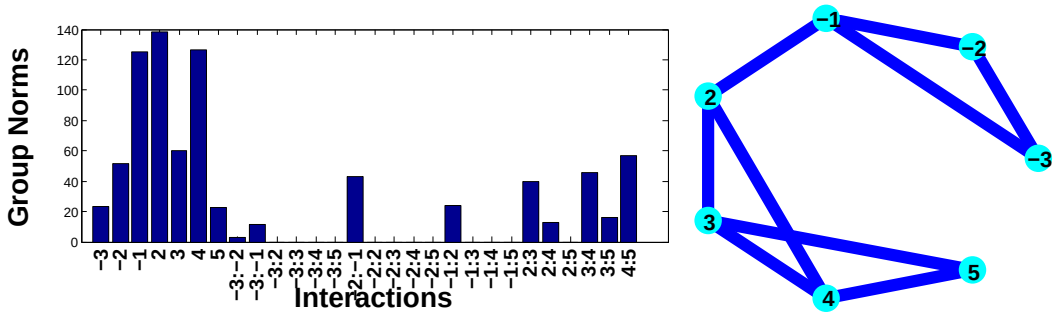


Figure 6.7.: **Top panel:** A bar graph representing the result of variable selection via simulated annealing. **Bottom panel:** An alternate graphical representation of the selected interactions based on the non-zero group norms. The circles indicate the individual variables and the lines indicate first-order (pairwise) interactions.

annealing extension of the framework. perform

6.7 Summary

In this chapter, we focused on the task of generating sparse point estimates of the regression coefficients which domain practitioners often utilize for interpreting the results of data analysis. We started with the goal of obtaining a truly sparse point estimate in a more principled way as opposed to heuristics like thresholding. We justified the choice of simulated annealing as a way to obtain this sparse estimate. Using simulated annealing, we showed how easily the existing framework of Gibbs sampling can be extended to produce a point estimate for the regression coefficients. Since the point estimate is generated with a changed version of the optimization problem (joint MAP over $\boldsymbol{\beta}$ and Λ), we provided

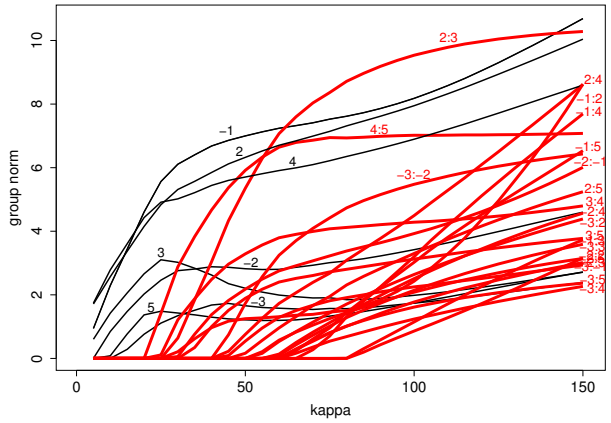


Figure 6.8.: Plot of the solution path based on a standard Group-Lasso penalized likelihood approach i.e. the traces of the group norms under relaxation of the constraint κ. The results lack more information about the posterior distribution of the regression coefficients.

a formal proof of why such an estimate will tend to be sparse. We showed how based on a slight alteration of the value of α, the solution will correspond directly to the MAP solution in the original MAP estimation problem. Extensions of the proof were discussed for both grouped variable selection and generalized linear models. In the next chapter, we conclude this thesis by providing a broader view of other parallel developments in the domain of Bayesian variable selection.

7 Conclusion

7.1 Bayesian Grouped Variable Selection

In the work described so far, we have constructed an omnibus Bayesian framework for grouped variable selection in the context of linear regression models. We started out with a framework for least squares regression (Gaussian likelihood) and showed how it can be extended further to generalized linear models. In particular, the binomial and the Poisson models were discussed with real-world experiments. Another modeling extension involved adding a clustering component, thereby performing grouped variable selection simultaneously for each cluster identified in the model. This extension was demonstrated in the context of survival analysis via a survival mixture-of-experts model. Based on these components, the final framework is capable of dealing with a variety of application scenarios.

The key motivation for using a Bayesian framework was to summarize the posterior distribution over the regression coefficients. We showed how the Bayesian grouped variable framework coupled with MCMC sampling for inference enabled us to generate estimates for the moments of the posterior distribution over the regression coefficients. The availability of such estimates is one of the main scoring points over the optimization based models. Additionally, we extended the Bayesian framework to generate a MAP estimate via simulated annealing. Hence the posterior distribution over regression coefficients can be summarized using the these estimates. A further flexibility with respect to grouped variable selection was introduced in the form of an extra sparsity tuning parameter. The extra parameter provided the framework with additional flexibility to control the level of sparsity in the solution without excessive shrinkage of the regression coefficients.

In view of our contributions with respect to the Bayesian framework for grouped variable selection, in this chapter, we discuss other parallel developments in the field of Bayesian variable selection. The idea is to get a broader view on Bayesian variable selection and evaluate further possibilities for improvement in this genre of methods. We break up this discussion into two parts. In the first part, we look at alternate constructions of sparsity inducing priors with a greater focus on the spike-and-slab prior. In the second part of the discussion, our focus is more on alternate inference mechanisms proposed for analyzing the posterior over the regression coefficients. The main focus in this part is on Bayesian variational approximation methods in the context of variable selection. Finally, we conclude with some future possibilities of this work.

7.2 Sparsity Inducing Prior Distributions

To simplify the discussion in this section, we mostly restrict our attention to the single variable selection case.

Scale Mixture of Normals. In this work, we have built our Bayesian framework for variable selection using a scale-mixture of normal distributions. The scale mixture of normals (see [18]), is a family of priors written as:

$$p(\beta_i) = \int N(\beta_i|0, \Psi_i)G(\Psi_i)d\Psi_i, \tag{7.1}$$

where G is a mixing distribution. In particular, in this work, we used normal-gamma class of priors:

$$p(\boldsymbol{\beta}|\bullet) \propto \int N(\boldsymbol{\beta}|\mathbf{0}, \sigma^2\Lambda) \prod_{g=1}^{d} \text{Gamma}(\lambda_g^2|\bullet)d\Lambda. \tag{7.2}$$

We saw that a specific case of this prior corresponds to the product of Laplacian distributions. The general class of normal-gamma priors and their properties are discussed in detail in [68]. It is possible to use other members of the scale mixture of normals family to build alternate sparsity inducing priors. An example of an alternate class of priors is the normal-inverse Gaussian prior as defined in [21]:

$$p(\boldsymbol{\beta}|\bullet) \propto \int N(\boldsymbol{\beta}|\mathbf{0}, \sigma^2\Lambda) \prod_{g=1}^{d} \text{Inv-Gaussian}(\lambda_g^2|\bullet)d\Lambda, \tag{7.3}$$

where "Inv-Gaussian" denotes the inverse-Gaussian distribution. In both these prior specifications, the sparsity inducing properties of the distribution over $\boldsymbol{\beta}$ depends on the way the mixing distribution is specified. We saw in earlier chapters how the value parameters of the gamma distribution in eq. (7.2) affected the sparsity inducing nature of the prior and also flexibility of the prior with respect to tuning the level of sparsity in the solution.

Broadly, the above classes of priors, which are part of the scale mixture of normals family, fall under the category of absolutely continuous priors where the mixture over the normals is continuous. We now look at an interesting alternate sparsity inducing prior, known as the spike-and-slab which is constructed based on a discrete mixing distribution.

Spike-and-Slab Prior. The foundations of spike-and-slab (SS) were laid out in [69], which defined a mutually independent prior for each component of $\boldsymbol{\beta}$. The initial version of the prior was defined as follows:

$$p(\beta_i = 0) = h_{0i}, \tag{7.4}$$

$$p(\beta_i < b, \beta_i \neq 0) = (b + f_i)h_{1i}, \quad -f_i < b < f_i, \tag{7.5}$$

and

$$p(|\beta_i| > f_i) = 0, \tag{7.6}$$

where $h_{0i} > 0$, $h_{1i} > 0$ and $h_{0i} + 2h_{1i}f_i = 1$. The construction is basically based on breaking up the prior distribution over β_i into two parts: a spike (at zero) and a uniform distribution (slab) in a range-bound interval. A point to note here is that unlike the other priors that we have discussed, this prior puts a non-zero mass at zero, which is represented here by the value h_{0i}.

In order to make posterior analysis feasible through MCMC sampling, latent variables were added later in [70] and [20] to create a hierarchical prior similar to the way we used scale mixture of normals. These versions had the distinction of not putting a probability mass on $\beta_i = 0$. The latent variable spike-and-slab model for linear regression with Gaussian likelihood, as defined in [20] is as follows:

$$\begin{aligned} y &\sim N(\boldsymbol{x}^t\boldsymbol{\beta}, \sigma^2) \\ \beta_i|\zeta_i, \tau_i^2 &\sim N(0, \zeta_i\tau_i^2) \\ \zeta_i|v_0, w &\sim (1-w)\delta_{v_0}(\cdot) + w\delta_1(\cdot) \\ \tau_i^{-2}|a_1, a_2 &\sim \text{Gamma}(a_1, a_2) \\ w &\sim \text{Uniform}[0, 1] \\ \sigma^2 &\sim \text{Gamma}(b_1, b_2). \end{aligned} \tag{7.7}$$

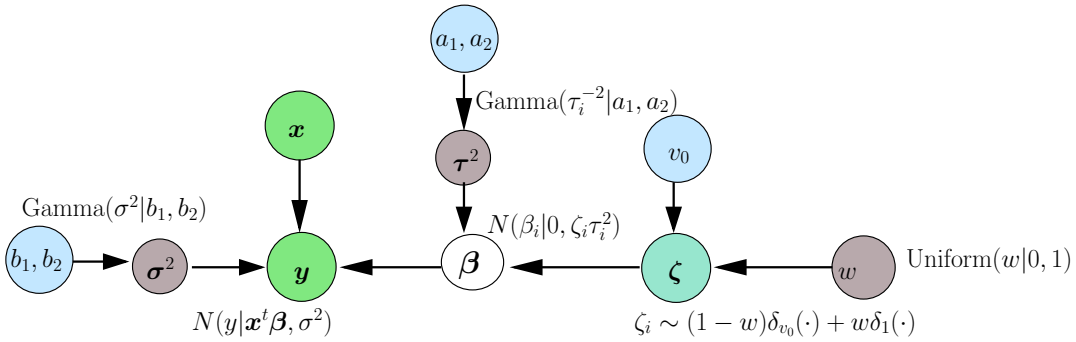


Figure 7.1.: The full hierarchical model for the spike-and-slab model for linear regression with auxiliary variables. The introduction of auxiliary variables makes posterior inference feasible via MCMC sampling since all the posterior conditional distributions are of standard form.

Based on this hierarchical prior, we now look at how model extensions that we carried out for our framework can be done for the spike-and-slab. The first extension involves grouped-variable selection. The spike-and-slab can be extended to grouped-variable selection by assigning a common ζ_g for each group of $\boldsymbol{\beta}_g$ coefficient vectors [71]. This ensures that the sparsity is at a group level. The second extension is regarding generalized linear models. Based on the hierarchical model in eq. (7.7), we can easily extend this model to generalized linear models in the same way as was done in chapter 4 by introducing a latent variable $\boldsymbol{\eta}$. Since posterior inference is not analytically feasible, approximate inference techniques

need to be used to estimate various quantities from the posterior distribution over the regression coefficients. The hierarchical prior specified in eq. (7.7) makes it convenient to use MCMC sampling. In particular, Gibbs sampling can be used for the above prior definition since all the conditionals are of standard form (see [20]). Regarding generating a MAP estimate, annealing can be tried as before for variables $\boldsymbol{\beta}, \boldsymbol{\zeta}$ and $\boldsymbol{\tau}$. However, it does not seem straightforward to provide a more theoretical insight into the properties of the MAP in this case (as was done in our framework). Further work is required to analyze the properties of the joint MAP estimate.

Hence so far, the spike-and-slab performs almost similarly to the normal-gamma prior that we have used in our framework. The computational costs for both inference algorithms is also similar with no added advantage of using the spike-and-slab. The Bayesian framework that we defined has an added advantage of generating a MAP estimate with an understanding of its properties. In the next section we will look at the inference aspect of sparse models for possible computational gains.

7.3 Bayesian Variational Methods for Variable Selection

After having discussed alternate sparsity inducing priors, we now look at the variable selection problem in a Bayesian regime from the perspective of the inference algorithm. In earlier chapters, we demonstrated the use of MCMC sampling for producing samples which resemble samples from the posterior distribution of the regression coefficients. Based on these samples, various quantities like expectation and variances were estimated. We saw the ease with which MCMC sampling can be implemented and the flexibility that it offers for incorporating various modeling extensions with minimal changes to the sampling algorithm. We now look at the possibility of using an alternate class of inference techniques in the Bayesian regime known as Bayesian variational methods.

Bayesian Variational Methods. In the Bayesian regime, another class of inference techniques are the Bayesian variational methods which use a different approach for approximating a target distribution as opposed to sampling. In a linear regression setting, our goal is to estimate various quantities from the posterior distribution over the regression coefficients:

$$p(\boldsymbol{\beta}|\boldsymbol{y}, X, \bullet) \propto p(\boldsymbol{y}|X\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}|\bullet). \tag{7.8}$$

Since the analysis of the posterior is not usually feasible analytically, an alternate approach is to approximate the posterior distribution with another distribution whose properties can be analyzed analytically. The key idea here is to translate this to an optimization problem where the goal is to find the "closest" distribution (among a chosen class of distributions) which is most similar to the posterior distribution of the regression coefficients. This involves minimizing the Kullback-Liebler (KL) divergence between the true posterior distribution and an approximated distribution, chosen from a specific class of distributions which are easier to analyze. Let p be the true posterior distribution and q represent a class of approximated distributions (say for example a multivariate Gaussian). The goal

is to then find a particular q distribution which comes closest to p, where the dissimilarity is measured via KL divergence defined as:

$$KL(q\|p) = -\int_{\boldsymbol{z}} q(\boldsymbol{z})\ln\frac{p(\boldsymbol{z}|\theta)}{q(\boldsymbol{z})}d\boldsymbol{z}. \tag{7.9}$$

Since KL divergence is not a symmetric divergence, the optimization can also be based on minimizing the reverse KL divergence:

$$KL(p\|q) = -\int_{\boldsymbol{z}} p(\boldsymbol{z}|\theta)\ln\frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\theta)}d\boldsymbol{z}. \tag{7.10}$$

Using eq. 7.9 leads to the method of Bayesian variational approximation and eq. 7.10 is used in the method known as expectation propagation (EP). In Bayesian variational approximation, the class of q distributions that is considered is usually a factorized set of distributions which are then optimized by iterating over the factors and optimizing each factor separately keeping the other factors fixed.

In expectation propagation, as described in [72], the general approach involves approximating q to a multivariate normal distribution and then making iterative improvements by approximating q to each factor in p, while keeping all the other factors fixed with the intention of matching the moments of the two distributions. In the context of variable selection in linear regression models, the use of expectation propagation for posterior inference has been demonstrated in [73]. The model uses a product of Laplacian distributions as prior over the regression coefficients (as used in the Bayesian Lasso). The Laplacian prior was used since it leads to a log-concave posterior and variational methods are more prone to be more robust in such cases. The inference algorithm was shown to be computationally efficient for large datasets. An extension to grouped-variable selection using a multivariate Laplacian prior has been used in [24] which also uses expectation propagation for posterior inference.

The inference algorithm mentioned above has shown to be computationally more efficient than its MCMC counterpart. As mentioned in [73], there is still the question of an approximation bias that exists with variational methods, whereas MCMC methods are not bound by such a bias. This means that sampling longer can improve MCMC estimates whereas the same is not possible with variational methods. It would be interesting to compare the estimates from both inference mechanisms to judge the qualitative aspect of the results. Also, with MCMC methods, it was possible to extend the variable selection model in different ways due to its flexibility. This was particularly useful in the case of clustering which was discussed in chapter 5. In the clustering case, the mixture-of-experts model resulted in a complicated hierarchical model which also included a distribution over the predictor variables. Inspite of these complicated extensions to the grouped-variable selection model, MCMC sampling was easily generalized to accommodate these extensions. In the case of EP, the same would have to be analyzed to see which of these extensions are feasible. This would be one of the possible directions for our future work.

7.4 Outlook

To sum up, in this work we have described a comprehensive Bayesian framework for grouped-variable selection with multiple advantages. The framework was extended to generalized linear models like the Poisson and binomial model and in principle it can be extended to other GLMs as well. Various estimates of the posterior distribution over the regression coefficients can be obtained from the resulting MCMC samples, like expectation and variance estimates. Simulated annealing was applied to obtain a MAP estimate with a formal justification of the validity of the estimate. The code used for this work, which is in C++ wrapped in R scripts, will soon be publicly available.

Looking ahead, we can think about the possibilities of further work based on our Bayesian framework for grouped variable selection. From a theoretical perspective, it would be useful to prove convergence formally for simulated annealing for the specific case of Gibbs sampling applied to variable selection. Although it is generally hard to prove convergence in continuous spaces, it may be interesting to evaluate the ideas in the work [61] and [62] to prove convergence. So far, in most experiments, we assumed that the data for all the experiments was complete, i.e. it did not contain any missing values. The only exception to this was in the case of survival analysis where censored observations were also included. Apart from that, any missing values for experiments were imputed in a pre-processing step. An extension to the framework could be to model the missing values by treating them as unknown parameters similar to the work in [74]. For computational gains it might also be worthwhile to investigate other inference techniques in the Bayesian regime. Expectation propagation for variable selection was briefly discussed in this chapter and a detailed comparison can be made between different inference techniques and what they have to offer. From an application perspective, a very interesting application of grouped-variable selection that can be pursued is the archetype analysis problem (see [75],[76]) which deals with identifying suitable archetypes from data assuming that the data is generated as a convex combination of the archetypes.

Appendices

A

Probability Distributions

Multivariate Normal Distribution

For a d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$, the multivariate normal or Gaussian distribution is defined as:

$$N(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{A.1}$$

where $\boldsymbol{\mu}$ is the expected value of $\boldsymbol{x}$ and Σ is the covariance matrix.

Gamma Distribution

For a positive scalar random variable, $x > 0$, the gamma distribution is defined as:

$$\text{Gamma}(x|a, b) = \frac{1}{\Gamma(a)} b^a x^{a-1} \exp(-bx), \tag{A.2}$$

where a and b are known as the shape and rate parameters respectively and Γ denotes the gamma function.

Inverse Gamma Distribution

For a positive scalar random variable, $x > 0$, the inverse gamma distribution is defined as:

$$\text{InvGamma}(x|a, b) = \frac{1}{\Gamma(a)} b^a x^{-a-1} \exp\left(-\frac{b}{x}\right), \tag{A.3}$$

where a and b are known as the shape and scale parameters respectively.

Poisson Distribution

For a non-negative integer $x = \{0, 1, 2, ...\}$, the Poisson distribution is defined as:

$$\text{Poisson}(x|\lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}. \tag{A.4}$$

Bernoulli Distribution

For a single binary variable $x \in \{0, 1\}$, the Bernoulli distribution is defined as:

$$\text{Bernoulli}(x|p) = p^x(1-p)^{1-x}, \tag{A.5}$$

where $p \in (0, 1)$.

Weibull Distribution

For a positive scalar random variable, $x > 0$, the Weibull distribution is defined as:

$$\text{Weibull}(x|\alpha, \lambda) = \frac{\alpha}{\lambda}\left(\frac{x}{\lambda}\right)^{\alpha-1} \exp(-(x/\lambda)^\alpha), \tag{A.6}$$

where α is the shape parameter and λ is the scale parameter.

Generalized Inverse Gaussian Distribution

For a positive scalar random variable, $x > 0$, the generalized inverse Gaussian distribution is defined as:

$$\text{GIG}(x|a, b, p) = \frac{(a/b)^{p/2}}{2K_p(\sqrt{ab})} x^{(p-1)} \exp\left(-(ax + b/x)/2\right), \tag{A.7}$$

where K_p is a modified Bessel function of the second kind, $a > 0$, $b > 0$ and p is a real parameter.

Dirichlet Distribution

The Dirichlet distribution is a multivariate distribution over K random variables $\mu_k \in [0, 1]$ where $k = \{1, ..., K\}$, subject to the constraint $\sum_{k=1}^K \mu_k = 1$ and is defined as:

$$\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\pi}, \alpha) = \frac{\Gamma(\alpha \sum_{k=1}^K \pi_k)}{\alpha^K \prod_{k=1}^K \Gamma(\pi_k)} \prod_{k=1}^K \mu_k^{\alpha\pi_k - 1}, \tag{A.8}$$

where $\boldsymbol{\mu} = (\mu_1, ..., \mu_K)$, $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$, $\sum_{k=1}^K \pi_k = 1$, $\alpha > 0$.

Exponential Distribution

For a scalar $x \geq 0$, the exponential distribution is defined as:

$$\text{Expd}(x|\lambda) = \lambda \exp(-\lambda x), \tag{A.9}$$

where $\lambda > 0$ is called the rate parameter.

Laplace Distribution

For $\boldsymbol{x} \in \mathbb{R}^d$, the multivariate Laplace distribution is defined as:

$$\text{M-Lap}(\boldsymbol{x}|\boldsymbol{\mu}, c) = \frac{1}{2c} \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{\mu}|}{c}\right), \tag{A.10}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the location parameter and $c > 0$ is the scale parameter. It is also sometimes called the double exponential distribution, because it can be viewed as two exponential distributions back to back.

B

Proportional Hazards and Accelerated Failure Time Models

In this section, we give details of how the proportional hazards model and the accelerated failure time model coincide for the Weibull distribution as shown in [47]. We begin by defining the hazard and survival function for the Weibull distribution:

$$\begin{aligned} h(t) &= \lambda_w^{\alpha_w} \alpha_w t^{\alpha_w - 1} \\ S(t) &= \exp(-\lambda_w t^{\alpha_w}), \end{aligned} \tag{B.1}$$

where α_w and λ_w are the shape and scale parameters of the Weibull distribution.

For modeling the effect of covariates, both the proportional hazards and the accelerated failure time models use $\exp(-\boldsymbol{x}^t\boldsymbol{\beta})$ in a multiplicative way. The difference is that in one, the hazard is multiplied by this term and in the other the time is rescaled by this term. Assuming this effect of the covariates on either the hazard or the time, the condition for equivalence of the two models can be stated as:

$$h_0(t)\exp(\boldsymbol{x}^t\boldsymbol{\beta}) = h_0^*(t\exp(\boldsymbol{x}^t\boldsymbol{\beta}^*))\exp(\boldsymbol{x}^t\boldsymbol{\beta}^*), \tag{B.2}$$

for all $\boldsymbol{x}$ and t. The LHS represents the proportional hazards model and the RHS represents the accelerated failure time model. Since this condition should hold for all $\boldsymbol{x}$ then it must also be true for the special case of $\boldsymbol{x} = \mathbf{0}$, where $\mathbf{0}$ is a vector of zeros. This implies that:

$$h_0(t) = h_0^*(t). \tag{B.3}$$

Hence, the two baseline hazards are the same in both models. To find the hazard function, consider a particular value of $\boldsymbol{x}$, where the first element of the vector is set to $-\log\left(\frac{t}{\beta_1^*}\right)$ and the others to zero:

$$\boldsymbol{x} = \left(-\log\left(\frac{t}{\beta_1^*}\right), 0, ..., 0\right). \tag{B.4}$$

Replacing this value of x in eq. (B.2) and simplifying we get:

$$h_0(t) = h_0(1)t^{\frac{\beta_1}{\beta_1^*}-1}. \tag{B.5}$$

We repeat the same procedure now with x having the value $-\log\left(\frac{t}{\beta_i^*}\right)$ in the i-th component and zero elsewhere. Again replacing this value of x in eq. (B.2) and simplifying we get:

$$h_0(t) = h_0(1)t^{\frac{\beta_i}{\beta_i^*}-1}. \tag{B.6}$$

Hence, since eq. (B.6) holds for all x, the ratios of the coefficients must be constant, i.e. $\frac{\beta_i}{\beta_i^*} = \alpha_w$. This leads to the hazard function:

$$h_0(t) = h_0(1)t^{\alpha_w-1}, \tag{B.7}$$

which matches the hazard function defined for the Weibull distribution in eq. (B.1), where $h_0(1) = \lambda_w^{\alpha_w}\alpha_w$. This result shows that the accelerated failure time and proportional hazards models coincide for the Weibull distribution.

Bibliography

[1] P. McCullaghand and J. Nelder, *Generalized Linear Models*. Chapman & Hall, 1983.

[2] L. Breiman, "Better subset regression using the non-negative garrote," *Technometrics*, vol. 37, pp. 373–384, November 1995.

[3] M. Yuan and Y. Lin, "On the non-negative garrote estimator," *Science*, vol. 30332, no. 2, pp. 143–161, 2005.

[4] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. pp. 267–288, 1996.

[5] I. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[6] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[8] M. Kyung, J. Gill, M. Ghosh, and G. Casella, "Penalized regression, standard errors, and Bayesian Lassos," *Bayesian Analysis*, vol. 5, no. 2, pp. 369–412, 2010.

[9] M. R. Osborne, B. Presnell, and B. A. Turlach, "On the Lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 319–337, 1999.

[10] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Stat. Soc. B*, pp. 49–67, 2006.

[12] V. Roth and B. Fischer, "The Group-Lasso for generalized linear models: uniqueness of solutions and efficient algorithms," in *ICML '08*, pp. 848–855, ACM, 2008.

[13] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused Lasso," *Journal of the Royal Statistical Society Series B*, pp. 91–108, 2005.

[14] N. Meinshausen, "Relaxed Lasso," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 374–393, 2007.

[15] H. Zou, "The adaptive Lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, pp. 1418–1429, Dec. 2006.

[16] D. Gamerman and H. F. Lopes, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition*. Chapman and Hall, 2006.

[17] T. Park and G. Casella, "The Bayesian Lasso," *Journal of the American Statistical Association*, vol. 103, pp. 681–686, June 2008.

[18] D. F. Andrews and C. L. Mallows, "Scale mixtures of normal distributions," *Journal of the Royal Statistical Society Series B Methodological*, vol. 36, no. 1, pp. 99–102, 1974.

[19] S. Raman and V. Roth, "Sparse Bayesian regression for grouped variables in generalized linear models," in *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pp. 242–251, Springer-Verlag, 2009.

[20] H. Ishwaran and J. S. Rao, "Spike and slab variable selection: Frequentist and Bayesian strategies," *ArXiv Mathematics e-prints*, May 2005.

[21] F. Caron and A. Doucet, "Sparse Bayesian nonparametric regression," in *ICML '08*, pp. 88–95, ACM, 2008.

[22] L. Meier, S. van de Geer, and P. Bühlmann, "The Group Lasso for Logistic Regression," *J. Roy. Stat. Soc. B*, vol. 70, no. 1, pp. 53–71, 2008.

[23] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simple MKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, November 2008.

[24] M. V. Gerven, B. Cseke, R. Oostenveld, and T. Heskes, "Bayesian source localization with the multivariate Laplace prior," *Advances in Neural Information Processing Systems*, pp. 1–9, 2009.

[25] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. Chapman&Hall, 1995.

[26] C. Hans, "Bayesian Lasso regression," *Biometrika*, vol. 96, no. 4, pp. 835–845, 2009.

[27] V. Seshadri, *The Inverse Gaussian Distribution: A Case Study in Exponential Families*. Oxford: Clarendon Press, 1993.

[28] D. Fink, "A compendium of conjugate priors. In progress report: Extension and enhancement of methods for setting data quality objectives.," *Technical Report*, 1995.

[29] B. Thompson, *Foundations of Behavioral Statistics: An Insight-based Approach*. Guilford Press, 1 ed., June 2008.

[30] C. Dahinden, G. Parmigiani, M. Emerick, and P. Bühlmann, "Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries," *BMC Bioinformatics*, vol. 8, p. 476, 2007.

[31] P. Green and T. Park, "Bayesian methods for contingency tables using Gibbs sampling," *Statistical Papers*, vol. 45, no. 1, pp. 33–50, 2004.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 1 ed., 2007.

[33] C. Perou, T. Sorlie, and Eisen, M.B. et al., "Molecular portraits of human breast tumours.," *Nature*, vol. 406, pp. 747–752, August 2000.

[34] D. Abd El-Rehim, G. Ball, and Pinder, S.E. et al., "High-throughput protein expression analysis using tissue microarray technology of a large well-characterised series identifies biologically distinct classes of breast cancer confirming recent cDNA expression analyses," *Int J Cancer*, vol. 116, pp. 340–50, 2005.

[35] R. Diallo-Danebrock, E. Ting, and Gluz, O. et al., "Protein expression profiling in high-risk breast cancer patients treated with high-dose or conventional dose-dense chemotherapy.," *Clin Cancer Res*, vol. 13, pp. 488–97, 2007.

[36] J. Kononen and Bubendorf, L. et al, "Tissue microarrays for high-throughput molecular profiling of tumor specimens.," *Nature Medicine*, vol. Jul;4(7), pp. 844–7, 1998.

[37] C. Elston and I. Ellis, "Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up," *Histopathology*, vol. 19, pp. 403–410, 1991.

[38] A. Raftery and S. Lewis, "One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo," *Statistical Science*, vol. 7, pp. 493–497, 1992.

[39] E. Dahl, G. K. G., and Gottlob, K. et al., "Molecular profiling of laser-microdissected matched tumor and normal breast tissue identifies karyopherin alpha2 as a potential novel prognostic marker in breast cancer.," *Clin Cancer Res*, vol. 12, pp. 3950–60, 2006.

[40] T. Sorlie, C. Perou, and Tibshirani, R. et al., "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications," *PNAS*, vol. 98, no. 19, pp. 10869–10874, 2001.

[41] S. G. Bottcher and C. Dethlefsen, "deal: A package for learning Bayesian networks," *Journal of Statistical Software*, vol. 8, pp. 200–3, 2003.

[42] S. G. Bøttcher and C. Dethlefsen., *deal: Learning Bayesian Networks with Mixed Variables*, 2009. R package version 1.2-33.

[43] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[44] G. Yeo and C. Burge, "Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals," *J. Comp. Biology*, vol. 11, pp. 377–394, 2004.

[45] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistics Society, Series B*, vol. 34, pp. 187–220, 1972.

[46] J. G. Ibrahim, M. H. Chen, and D. Sinha, *Bayesian Survival Analysis*. Springer-Verlag:New York Inc, 2001.

[47] D. R. Cox and D. Oakes, *Analysis of Survival Data*. Chapman and Hall, 1984.

[48] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," in *Advances in Neural Information Processing Systems 14*, pp. 881–888, MIT Press, 2002.

[49] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[50] S. Kim, P. Smyth, and H. Stern, "A nonparametric Bayesian approach to detecting spatial activation patterns in fMRI data," in *Proceedings of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention*, pp. 217–224, 2006.

[51] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 249–265, 2000.

[52] O. Rosen and M. Tanner, "Mixtures of proportional hazards regression models," *Statistics in Medicine*, vol. 18, pp. 1119–1131, 1999.

[53] T. Ando, S. Imoto, and S. Miyano, "Kernel mixture survival models for identifying cancer subtypes, predicting patient's cancer types and survival probabilities," *Genome Informatics*, vol. 15, no. 2, pp. 201–210, 2004.

[54] A. Kottas, "Nonparametric Bayesian survival analysis using mixtures of Weibull distributions," *Journal of Statistical Planning and Inference*, vol. 136, no. 3, pp. 578 – 596, 2006.

[55] J. G. Ibrahim, M. H. Chen, and S. N. Maceachern, "Bayesian variable selection for proportional hazards models," *The Canadian Journal of Statistics*, vol. 27, no. 4, pp. 701–717, 1999.

[56] M. D. Paserman, "Bayesian inference for duration data with unobserved and unknown heterogeneity: Monte Carlo evidence and an application," IZA Discussion Papers 996, Institute for the Study of Labor (IZA), Jan. 2004.

[57] H. Ishwaran and M. Zarepour, "Exact and approximate sum-representations for the Dirichlet process," *The Canadian Journal of Statistics,*, vol. 30, pp. 269–283, 2002.

[58] R. B. Gramacy and N. G. Polson, "Simulation-based regularized logistic regression," *ArXiv e-prints*, May 2010.

[59] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, 4598, pp. 671–680, 1983.

[60] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, January 1985.

[61] S. B. Gelfand and S. K. Mitter, "Metropolis-type annealing algorithms for global optimization in Rd," *SIAM J. Control Optim.*, vol. 31, pp. 111–131, January 1993.

[62] C. Andrieu, L. A. Breyer, and A. Doucet, "Convergence of simulated annealing using Foster-Lyapunov criteria.," *J. Appl. Probab.*, vol. 38, no. 4, pp. 975–994, 2001.

[63] G. Royer, "A remark on simulated annealing of diffusion processes," *SIAM Journal on Control and Optimization*, vol. 27, no. 6, pp. 1403–1408, 1989.

[64] L. Goldstein, "Mean square rates of convergence in the continuous time simulated annealing algorithm on Rd," *Adv. Appl. Math.*, vol. 9, pp. 35–39, March 1988.

[65] G. O. Roberts and O. Stramer, "Langevin diffusions and Metropolis-Hastings algorithms," *Methodology and Computing in Applied Probability*, vol. 4, pp. 337–357, 2002.

[66] H. Sagan, *Introduction to the calculus of variations*. 1992. Corrected reprint of the 1969 original.

[67] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, November 1999.

[68] J. E. Griffin and P. J. Brown, "Inference with normal-gamma prior distributions in regression problems," *Bayesian Analysis*, vol. 5, no. 1, pp. 171–188, 2010.

[69] T. J. Mitchell and J. J. Beauchamp, "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. pp. 1023–1032, 1988.

[70] E. I. George and R. E. McCulloch, "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[71] D. Hernández-Lobato, J. M. Hernández-Lobato, T. Helleputte, and P. Dupont, "Expectation propagation for Bayesian multi-task feature selection," *Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, vol. 6321, pp. 522–537, 2010.

[72] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *UAI*, pp. 362–369, 2001.

[73] M. Seeger, "Bayesian inference and optimal design in the sparse linear model," *Journal of Machine Learning Research*, vol. 9, pp. 759–813, 2008.

[74] A. Dobra, C. Tebaldi, and M. West, "Bayesian inference in incomplete multi-way tables," *Journal of Statistical Planning and Inference*, 2003.

[75] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, 1994.

[76] M. J. Palmer and G. B. Douglas, "A Bayesian statistical model for end member analysis of sediment geochemistry, incorporating spatial dependences," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 57, no. 3, pp. 313–327, 2008.